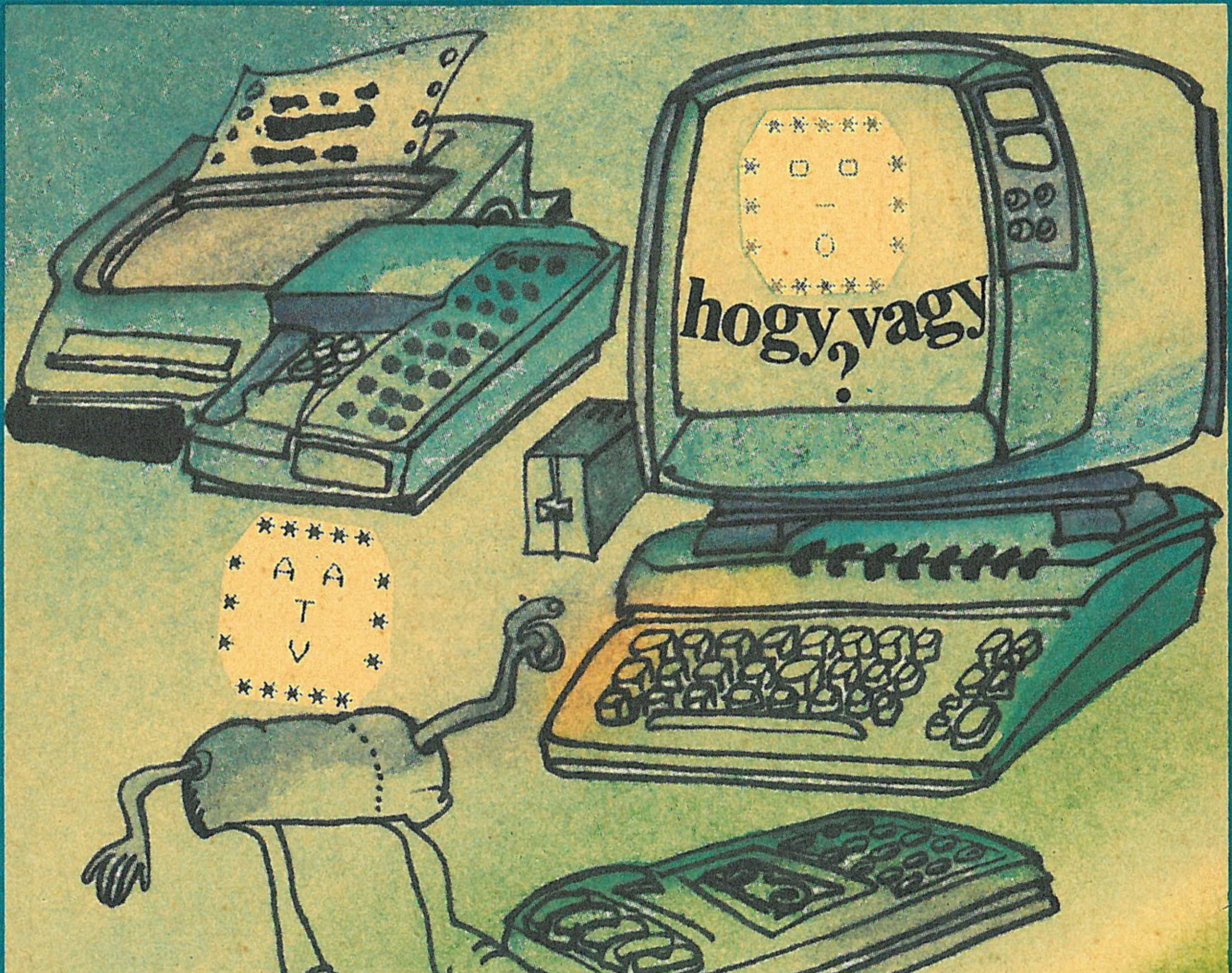
MAJOR ZOLTÁN
VALOVICS ISTVÁN

A BASIC FELADATOK TÜKRÉBEN



PÁRBESZÉD A SZÁMÍTÓGÉPPEL

MAJOR ZOLTÁN VALOVICS ISTVÁN

A BASIC feladatok tükrében

(Párbeszéd a számítógéppel)

TANKÖNYVKIADÓ, 1986

A könyv szerzői:

DR. MAJOR ZOLTÁN
egyetemi adjunktus

DR. VALOVICS ISTVÁN
gyakorlógimnáziumi tanár

Bírálók:

DR. HÁMORI MIKLÓS
kandidátus

LŐCS GYULA
okleveles alkalmazott
matematikus

Rajzolta:

SAJDIK FERENC

ISBN 963 17 8837 7

Szüleinknek

Előszó

Ezt a könyvet mindazoknak szánjuk, akik a BASIC programozási nyelvvel szeretnének megismerkedni. Ennek érdekében az egyes fejezetek didaktikusan épülnek egymásra, a fejezeteken belül pedig különböző feladatok megoldásával mélyíthetők el az ismeretek. Ezek alapos elsajátítását megkönnyíti, és igazából lehetővé csak az teszi, ha a tanultakat számítógépen ki is próbáljuk. Javasoljuk tehát az Olvasónak, hogy

- a könyv anyagának feldolgozását a személyi számítógéphez ülve kezdje el,
- próbálja ki az összes bemutatott példaprogramot,
- oldja meg az összes feladatot,
- oldjon meg minél több gyakorlatot.

Könyvünk olvasásához *nem szükséges különösebb matematikai előismeret.* Ebből a szempontból az ötödik fejezet jelenthet kisebb nehézséget, de az abban leírtak teljes megértését a további fejezetek nem igénylik.

A könyvben levő sok feladat és gyakorlat, amelyek mindegyikének a megoldását is közöljük, lehetővé teszi, hogy *tanári kézikönyvként is használható* legyen. A feladatok megoldásai közvetlenül az egyes fejezetek végén, a gyakorlatoké a tizenkettedik fejezetben vannak összegyűjtve. A gyakorlatoknál gondoljuk végig, hogy az adott fejezetben tanult ismeretek közül melyek segítségével tudjuk megoldani, vagy egyszerűbben megoldani azokat. Ismétlésnek is nagyon jó, ha minden egyes új ismerethez megkeressük azokat a gyakorlatokat – a korábbi fejezetekből is –, amelyek megoldását lerövidíthetjük, tökéletesíthetjük segítségükkel.

Ne keseredjünk azonban el, ha az általunk készített megoldás a közölttől különbözik, hiszen az is csak egy a lehetségesek közül. A feladatok és a gyakorlatok között olyanokkal is találkozunk, amelyeket többféleképpen is meg lehet oldani. Minden esetben tanulmányozzuk át a közölt programokat is, de saját programunk helyessége ellenőrzésének legbiztosabb módja, ha a számítógépen lefuttatjuk, és valóban azt csinálja, amire gondoltunk.

A programozási nyelvek annyiban hasonlítanak a természetes nyelvekhez, hogy szavakból és jelekből állnak, amelyeket nyelvtani szabályok szerint kapcsolunk össze. Alapvető különbség, hogy a programozási nyelvek szókincse lényegesen kisebb, és a helyesírási, formai szabályok jóval szigorúbbak. A számítógép csak akkor érti meg az általunk közölni kívánt információt, ha azt egyértelműen, a szabályoknak megfelelően írjuk le. Ha egy programot a formai szabályokat betartva, azaz szintaktikusan helyesen megírunk, ez még nem jelenti azt, hogy jó is lesz. Fontos hogy logikailag, azaz szemantikusan is helyes legyen.

Tartsuk szem előtt a következő jótanácsot!
Ha egy program nem úgy működik, ahogy azt szerettük volna, akkor ne a gépben, hanem programunkban keressük a hibát! Ennek megtalálása sokszor nem könnyű feladat. Ahhoz, hogy gyorsan és pontosan tudjunk programot írni, illetve a hibákat hamar megtaláljuk, nagyon sok gyakorlásra van szükség.

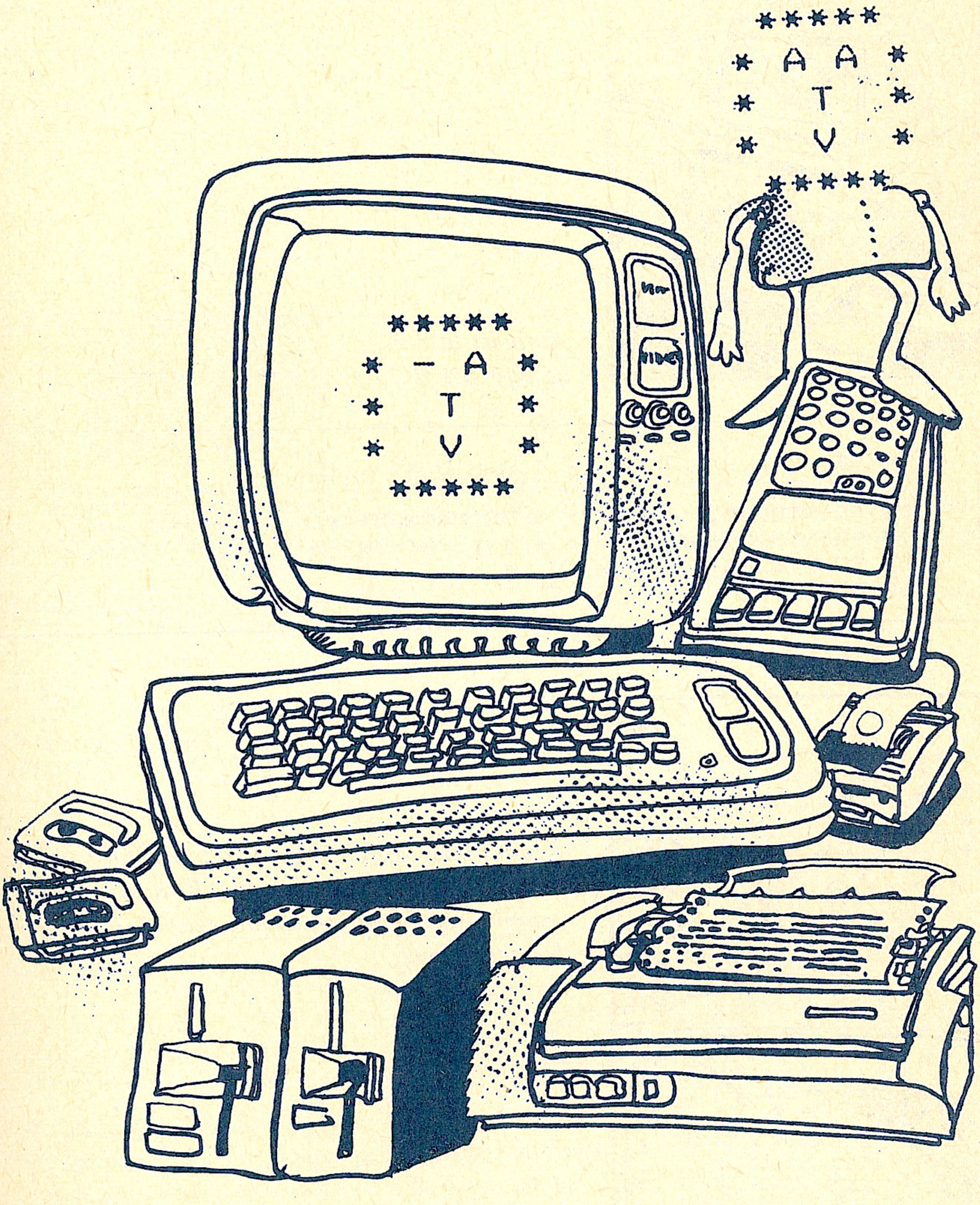
A könyvben a **HT iskolaszámítógép** BASIC nyelvének alapjait mutatjuk be. Más személyi számítógépek által ismert BASIC nyelvek több-kevesebb eltérést mutatnak, amelyek a gépekhez adott leírások alapján könnyen kideríthetőek, és ezeknek megfelelően a könyv tartalma, feladat- és gyakorlatanyaga ezekhez a számítógépekhez is jól használható.

Itt szeretnénk köszönetet mondani:
- Hámori Miklósnak és Lőcs Gyulának, könyvünk lektorainak, a kézirat alapos áttanulmányozásáért és értékes javaslataikért;
- a Híradástechnikai Szövetkezetnek, hogy lehetővé tette a programlisták kinyomtatását;
- Bazsó Júliának és Gyapjas Ferencnek a kézirattal kapcsolatos hasznos észrevételeikért;
- és mindazoknak, akik a könyv megjelenését elősegítették.

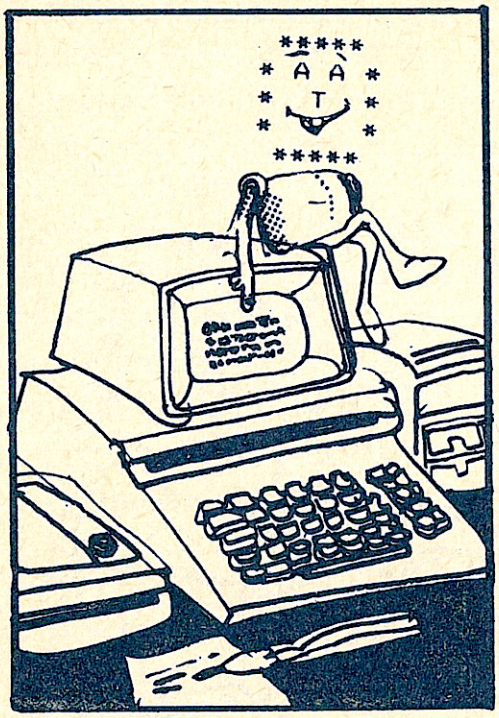
Ezek után még egy jótanács az Olvasónak:
A PROGRAMOZÁSHOZ SEM VEZET KIRÁLYI ÚT!



Bevezető

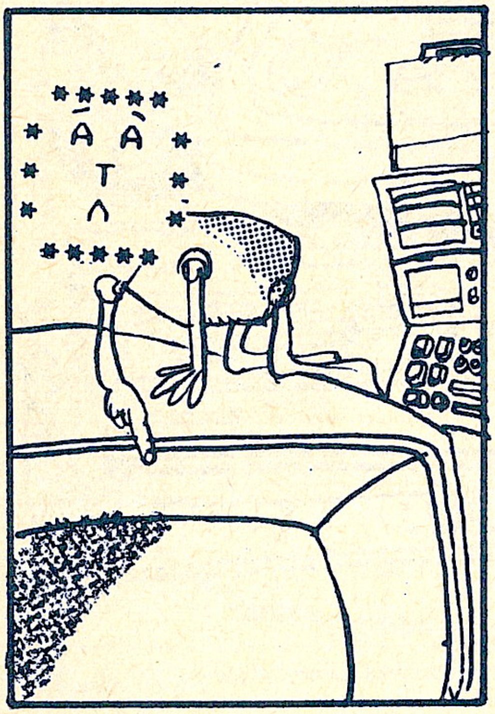


Vegyünk egy mély lélegzetet, és lapozzunk!



Amit itt látunk,
az egy BASIC nyelven írt program.

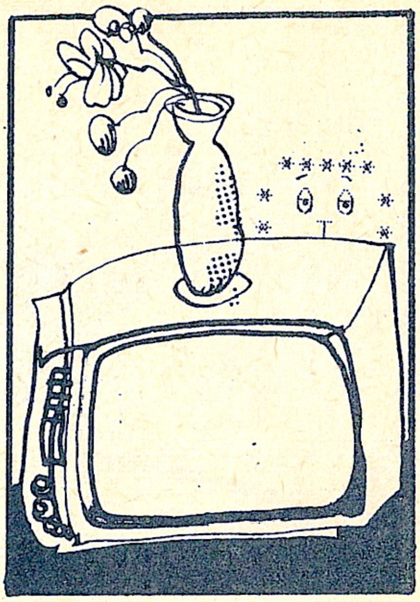
```
10 REM K I S E R O   F I G U R A
20 CLS
30 PRINT@285," *****"; : PRINT@285+4*64," *****";
40 PRINT@285+64,"* A A *"; : PRINT@285+2*64,"*  T  *";
50 PRINT@285+3*64,"*  V  *";
60 GOTO 60
```

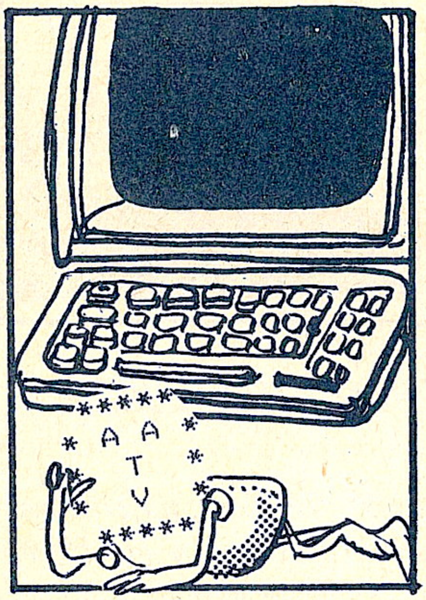


Önmagában nem sokat ér!

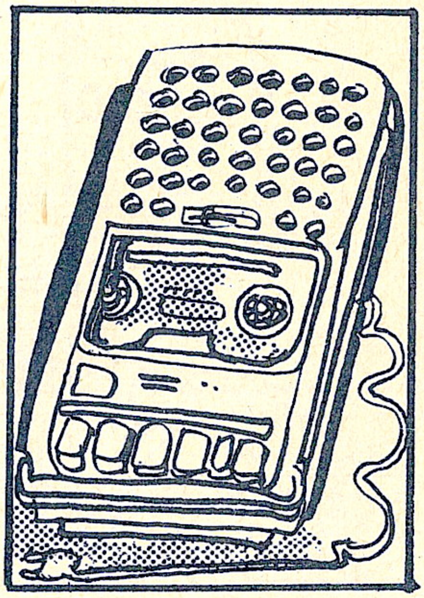
Számunkra érdekessé, felhasználhatóvá azután válhat, ha megismerkedünk a személyi számítógépek legfontosabb részeivel és azok használatával:

A képernyő:
A számítógépnek küldött és a számítógép által adott üzenetek képszerű megjelenítésére szolgál.





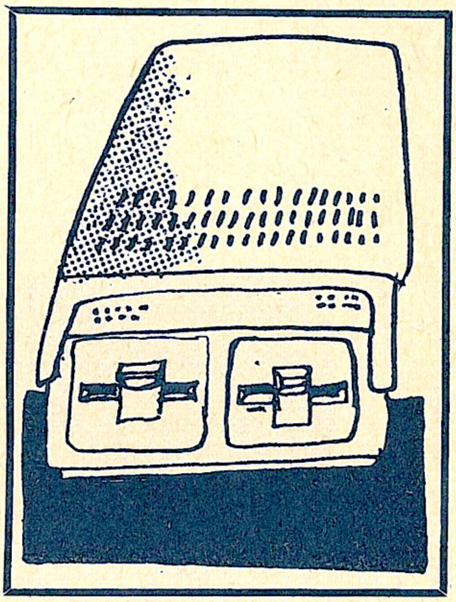
A központi egység és a billentyűzet:
A központi egység értelmezi és feldolgozza a billentyűzetről, a kazettás tárolóegységről, ... érkező utasításokat, parancsokat és jeleket.



A kazettás tárolóegység:
Ez lehet például egy hagyományos kazettás magnetofon, amelynek segítségével a programokat és az adatokat tudjuk magnószalagon tárolni.

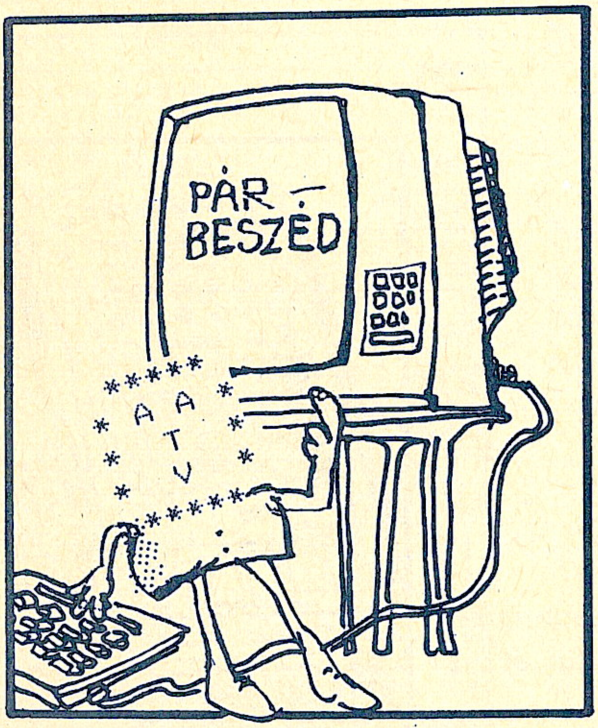


A sornyomtató:
Az információk papíron való rögzítésére szolgál.



A hajlékony mágneslemez egység:
Ugyanarra használják, mint a kazettás tárolóegységet, de annál gyorsabb és megbízhatóbb.

A képernyő és a billentyűzet segítségével a számítógéppel párbeszédet folytathatunk.



Ezt akkor tudjuk elkezdeni, ha a számítógépet üzembe helyezzük. A HT1Ø8ØZ iskola-számítógép képernyőjén a bal felső sarokban megjelenik a

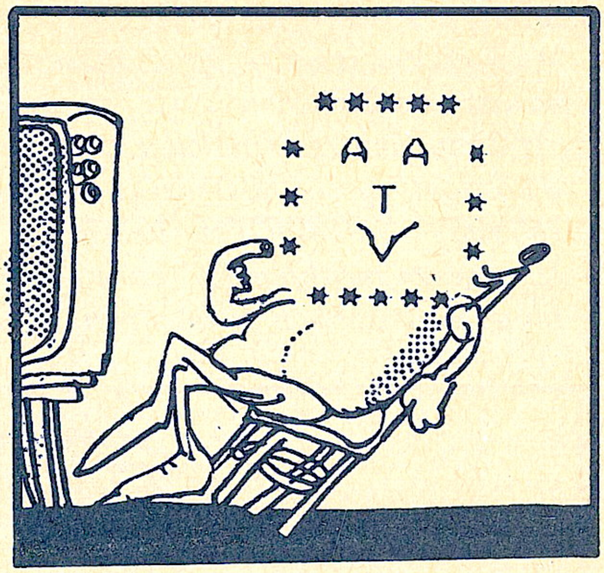
READY?_

felirat. Válaszul a NEWLINE *billentyűt kell lenyomni. Ezt követően a képernyő bal alsó sarkában az alábbi választ kapjuk:*

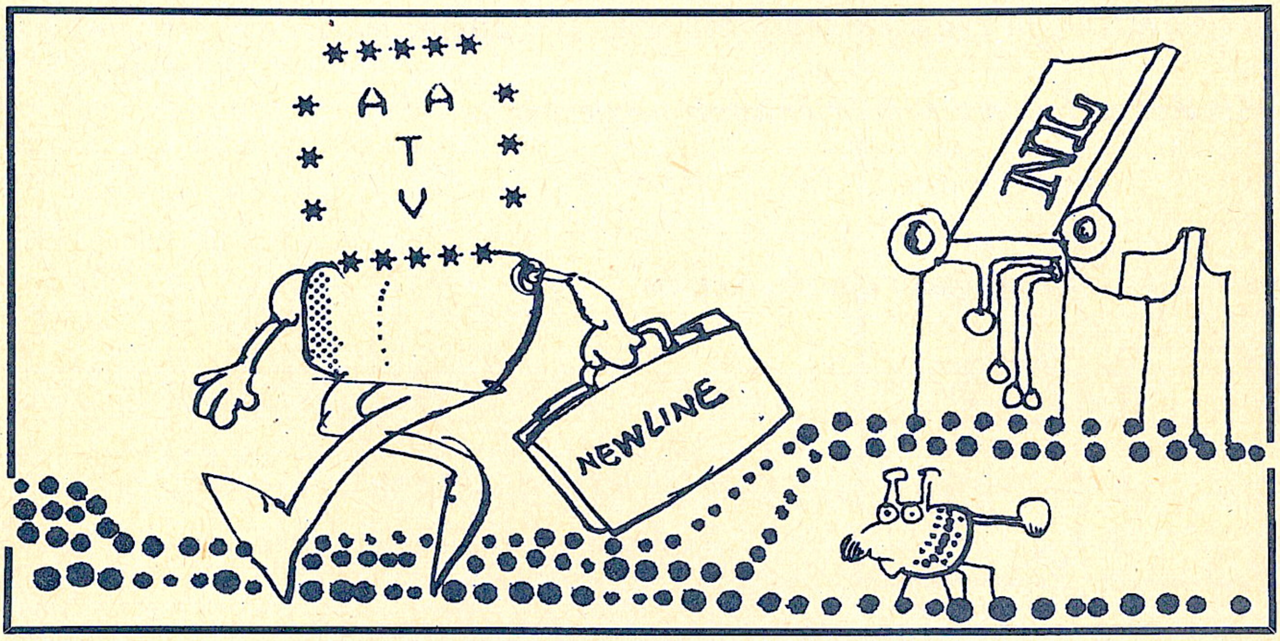
READY
> _

Ez mutatja, hogy a számítógép munkára kész.

Most rajtunk a sor!

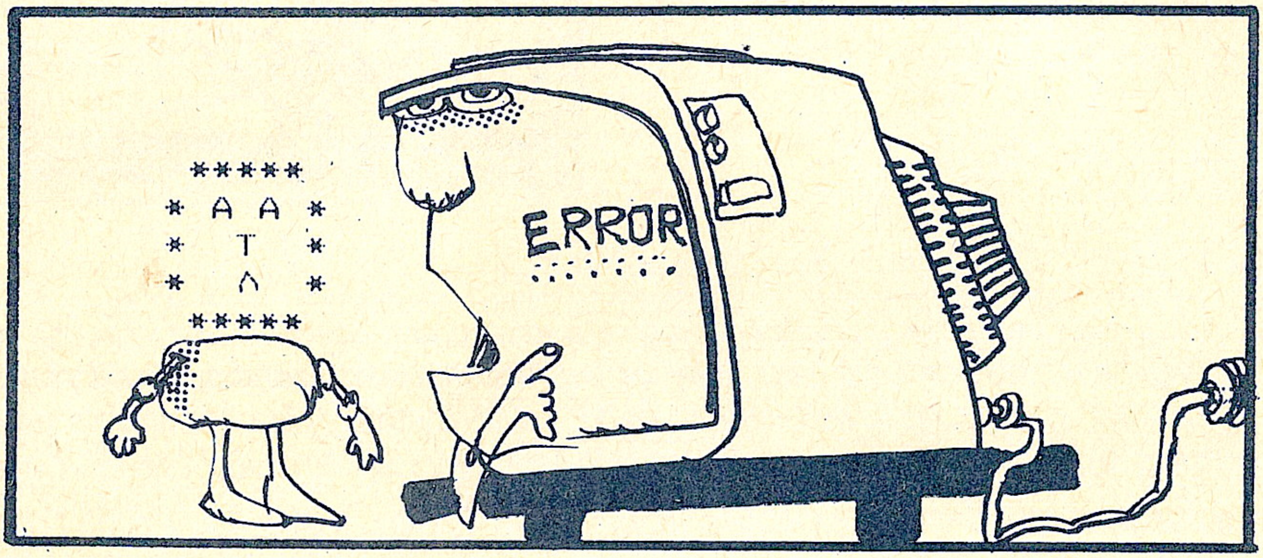


A billentyűzeten begépelhetünk valamilyen üzenetet, ami egyidejűleg a képernyőn is megjelenik. Ahhoz azonban, hogy ez az információ a számítógép központi egységéhez is eljusson, le kell nyomni a NEWLINE *billentyűt. A továbbiakban ennek a billentyűnek a lenyomását* NL*-lel jelöljük.*



Miután az üzenetet elküldtük, két eset lehetséges:

- *Az üzenet a számítógép számára értelmes, ekkor a központi egység értelmezi azt és végrehajtja.*
- *Az üzenetünk értelmetlen a számítógép számára, ekkor annak feldolgozásakor a központi egység hibajelzéssel (ERROR) válaszol. A hibás üzenet újragépeléssel vagy javítással (szerkesztéssel, EDIT) módosítható. A további munkát lényegesen megkönnyíti, ha a gépkönyvekből a szerkesztésre vonatkozó részt minél alaposabban áttanulmányozzuk.*



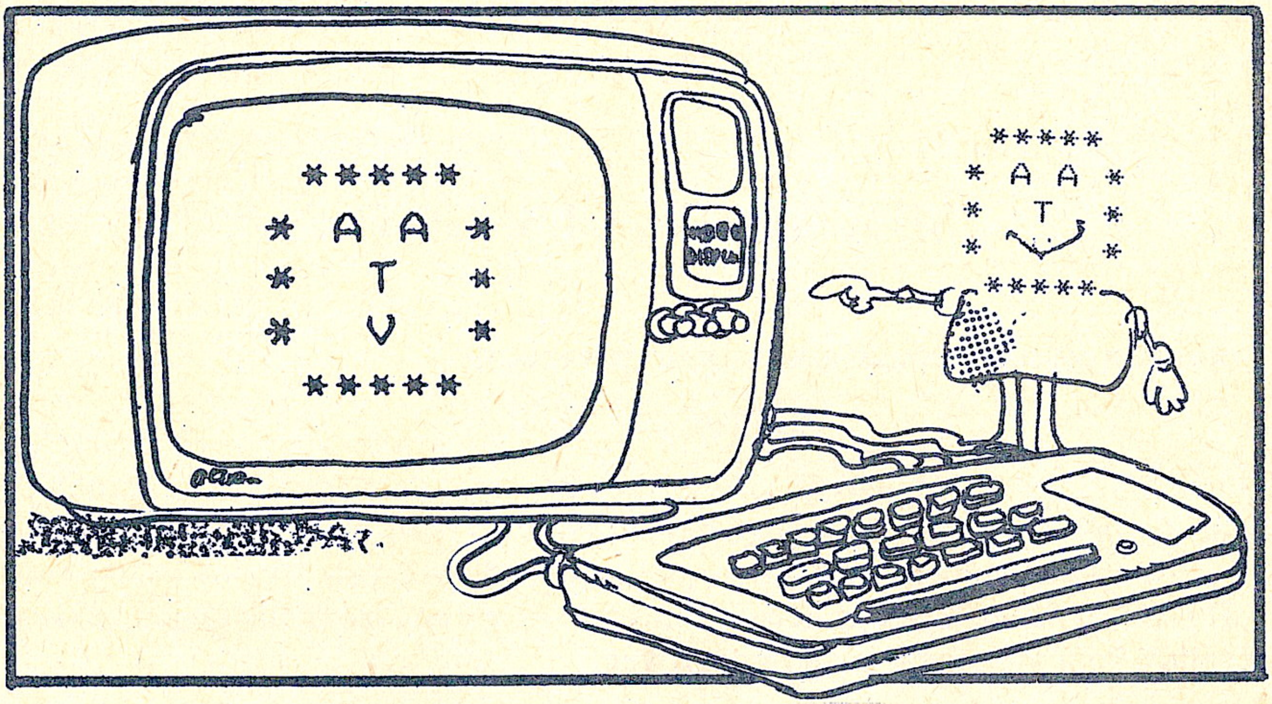
Jól jegyezzük meg: a számítógép nem szereti a „zöldségeket”!

Mindkét eset után a képernyőn ismét megjelenik a

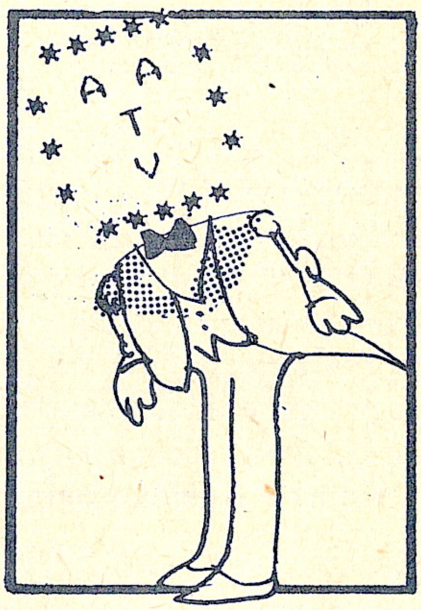
READY
> _

felirat, és ismét üzenetet gépelhetünk be.

Most már visszatérhetünk a 8. oldalon levő programhoz! Ha azt soronként begépeljük (ne feledkezzünk meg a sor végén a NL *-ről!), és a számítógépnek a* RUN NL *parancsot adjuk, akkor ennek hatására a képernyőn könyvünk kisérő figurája jelenik meg.*



A továbbiakban a BASIC nyelvű programozás alapjait mutatjuk be.



Első fejezet

ISMERKEDJÜNK A KÉPERNYŐVEL!

SET, RESET, CLS, GOTO

Adjuk a számítógépnek a következő parancsot!

SET(63,23) [NL]

Ennek hatására a képernyő 63. oszlopának 23. sorában megjelenik egy grafikus pont. *Grafikus pont*nak, vagy egyszerűen csak pontnak fogjuk nevezni az egy SET utasítással vagy paranccsal megjeleníthető képelemet.

Ha azt akarjuk, hogy a képernyőn csak ez az egy pont világítson, akkor gépeljük be a következő programsorokat:

1.1 példaprogram

```
10 CLS
20 SET(63,23)
30 GOTO 30
```

Ha most a RUN [NL] paranccsal elindítjuk ezt a programot, akkor a 10-es sorban levő **CLS** utasítás hatására minden törlődik a képernyőről, a 20-as sorban levő utasítás ismét kigyújtja az előző pontot. A 30-as sorban levő **GOTO 30** utasítással pedig elérhetjük azt, hogy a képernyőn a ponton kívül semmi sem jelenik meg. A gép ezt a sort hajtja végre mindaddig ismétlődve, amíg a programot a [BREAK] billentyűvel vagy a [RESET] gombbal le nem állítjuk, vagy a gépet ki nem kapcsoljuk. Eközben nem ír ki semmit a képernyőre, tehát csak a 20-as sorban kigyújtott pont világít.

Ha az 1.1 példaprogramban a 30-as sor helyett a következőket írjuk:

30 GOTO 10

azaz azt az utasítást adjuk a gépnek, hogy menjen vissza a 10-es sorra, akkor a számítógép állandóan ismételni fogja az egymást követő 10-es, 20-as és 30-as sorban levő utasításokat: törli a képernyőt, kigyújtja az adott pontot ... Ezzel tehát elértük, hogy az adott pont villog.

Egy pontot úgy is lehet villogtatni, hogy nem az egész képernyőt töröljük,

hanem csak a kigyújtott pontot a **RESET** utasítással. Az ennek megfelelő program pedig:

1.2 példaprogram

```
10 CLS
20 SET(63,23)
30 RESET(63,23)
40 GOTO 20
```

vagy ami ezzel egyenértékű:

1.3 példaprogram

```
10 CLS
20 SET(63,23) : RESET(63,23) :
   GOTO 20
```

Lehetőség van tehát arra, hogy az utasításokat egymástól kettősponttal elválasztva egy programsorba, illetve az egy programsorhoz tartozó utasításokat több sorba írjuk.

Megjegyzés: Ha már egy programra nincs szükségünk, akkor az új program begépelése előtt adjuk ki a **NEW** [NL] parancsot!

1.1 feladat: Készítsünk olyan programot, amelyik egy négyzet négy csúcsában és a középpontjában kigyújt egy-egy pontot!

1.2 feladat: Módosítsuk az előző programot úgy, hogy a négyzet csúcsaiban levő pontok villogjanak, a középpont pedig folyamatosan világítson!

1.3 feladat: Készítsünk olyan programot, amelyik kilenc középpontosan szimmetrikusan elhelyezkedő pontot gyújt ki a képernyőn! (Az összes pont ne legyen egy egyenesen!)

Értékadás

A SET és a RESET utasítások felhasználásával mozgatni is tudjuk a pontokat a képernyőn.

1.4 példaprogram

```
10 CLS
20 X=2
30 SET(X,23) : RESET(X,23) :
   X=X+1 :
   GOTO 30
```

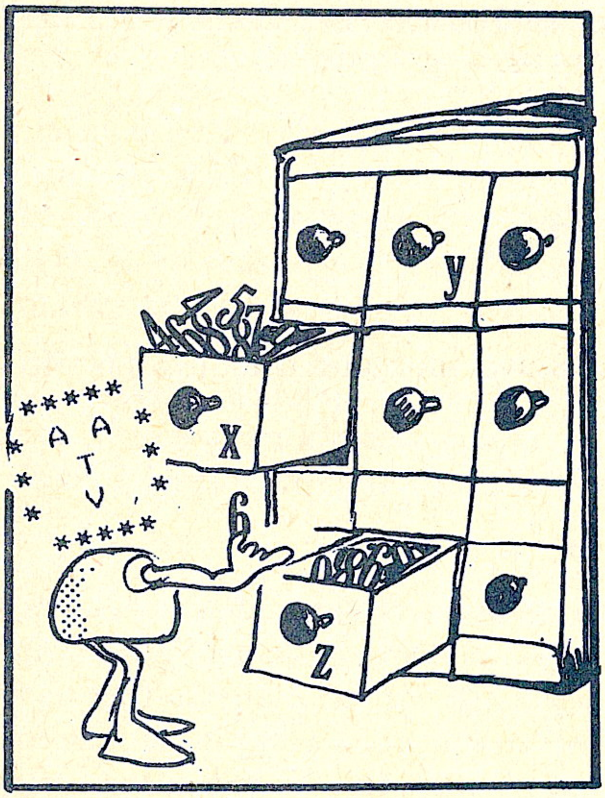
A példaprogramban az X változó segítségével a kigyújtandó pontok első koordinátáját lehet változtatni. A 2Ø-as sorban levő X=2 és a 3Ø-as sorban levő X=X+1 utasítások ún. *értékadó utasítás*ok. Ezek pontos alakja:
LET X=2 (azaz legyen az X változó értéke 2),
LET X=X+1 (azaz legyen az X változó értéke az eddigi értékénél 1-gyel nagyobb.) Ügyeljünk arra, hogy az itt szereplő egyenlőségjel jelentése az, hogy a bal oldalán álló változó értéke legyen egyenlő a jobb oldal értékével. A bal oldalon mindig csak egy változó állhat, a jobb oldalon viszont szerepelhet vagy egy változó, vagy egy állandó, vagy egy összetett kifejezés is. Az értékadó utasításokat a LET kulcsszó nélkül is felismeri a gép, így ez el is hagyható.

A GOTO 3Ø utasítás hatására ismét a 3Ø-as sor elejére kerül a program végrehajtása, tehát kigyullad a 23. sorban a következő pont, majd el is alszik, ... és így tovább.

1.5 példaprogram

```
10 CLS
20 X=120 : Y=1
30 SET(X,23) : RESET(X,23) :
   X=X-Y :
   GOTO 30
```

Ez a példaprogram az 1.4 példaprogramhoz hasonlóan egy pontot mozgat, de az ottanihoz képest ellenkező irányban.



Egyúttal az is látható, hogy lehetőség van változók értékeinek összeadására, kivonására. A 3Ø-as sorban az X=X–Y utasítás hatására az X változó értékéből a számítógép kivonja az Y változó értékét, és az X változó új értéke a kivonás eredménye lesz, tehát korábbi értékét elveszti. (Y értéke nem változik.) Előfordulhat, hogy az X változó előző tartalmát szeretnénk megőrizni. Ezt például úgy tehetjük meg, hogy a fenti utasítás elé beírjuk a Z=X értékadó utasítást, és ekkor a Z változó megőrzi a kívánt értéket.

Megjegyzés: A programokban a változók nevét kötelező betűvel kezdeni, a betű után írhatunk betűket vagy számokat is. Példák változónevekre:

A, B2, FLAG, VEG, ERTEK

A számítógép a változókat a nevük első két karaktere alapján különbözteti meg egymástól. Így például a SZAM és a SZEG nevek a gép számára ugyanazt a változót azonosítják.
A változónevek nem tartalmazhatják a BASIC kulcsszavakat. Ennek megfelelően a következő változónevek hibásak:

KELET	a LET kulcsszó miatt,
TOL	a TO kulcsszó miatt,
UGRUNK	a RUN kulcsszó miatt.

1.4 feladat: Írjunk olyan programot, amelyik egyszerre két pontot mozgat, az egyiket vízszintesen, a másikat függőlegesen!

Megjegyzés: A program futásánál előforduló FC Error hibaüzenettel kapcsolatban lásd a Grafikus képernyőméret című részt!

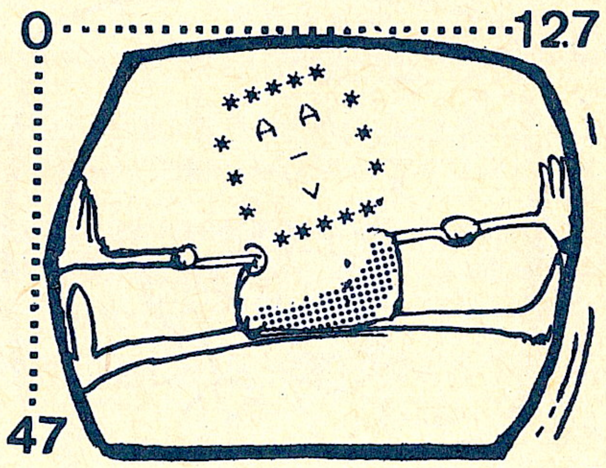
1.5 feladat: Próbáljuk meg az 1.4 példaprogramot úgy módosítani, hogy a pont folyamatosan fusson végig a 23. soron!

Grafikus képernyőméret

Az 1.4 és az 1.5 példaprogramok lefuttatáskor úgy fejeződnek be, hogy a számítógép hibajelzést ad. Ez nem véletlen, hiszen a képernyő méretei végesek, így a SET és a RESET utasítások argumentumába (a zárójelek közé) nem kerülhetnek akármilyen számok. A SET(X, Y), illetőleg a RESET(X, Y) utasítás csak akkor értelmes üzenet a gép számára, ha az X és az Y változók értékére teljesülnek a következő feltételek:

$$\emptyset \leqq X \leqq 127 \quad \text{és} \quad \emptyset \leqq Y \leqq 47$$

Ez azt jelenti, hogy a képernyőn 48 sorban soronként 128 pontot, vagyis összesen 6144 grafikus pontot tudunk megjeleníteni.



Megjegyzés: A 128 oszlop és a 48 sor számozása nullától kezdődik, ennek megfelelően a képernyő legfelső sora a nulladik sor, a bal szélső oszlopa a nulladik oszlop.

Több lehetőség is van arra, hogy az előbb említett hibát elkerüljük. Ezekkel a későbbiek során ismerkedünk majd meg.

Parancs, utasítás, program, programsor

Az alcímben szereplő fogalmak mindegyikével találkoztunk már, most pontosítsuk ezek jelentését.

A *parancs* a számítógép számára az az értelmes üzenet, amelyet a tartalmának megfelelően azonnal végre is hajt. Például kigyújt egy grafikus pontot (SET(12,15) [NL]), elindít egy programot (RUN [NL]), letörli a képernyőt (CLS [NL]) stb.

Az *utasítás* olyan értelmes üzenet, amelyet a számítógép nem közvetlenül a begépelés és az elküldés ([NL]) után hajt végre (hanem csak akkor, amikor a program futása az adott utasításhoz ér) és javításig vagy törlésig tárol.

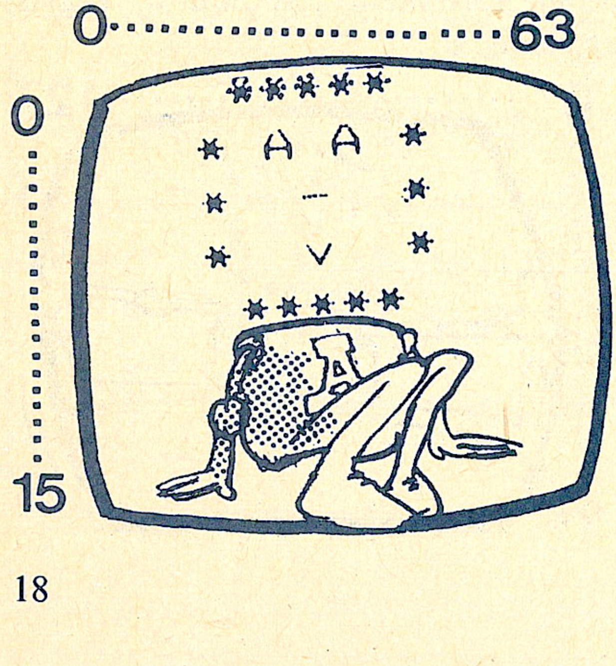
A *programsor* egy sorszámból (előjel nélküli egész szám) és egy vagy több utasításból áll. A sorszám utáni utasítások több sorba is kerülhetnek (például az 1.4 példaprogram 3Ø-as sorában három sorba).

A *program* programsorokból áll. A könnyebb javítás érdekében célszerű a programsorokat tízesével számozni. A számítógép ezeket növekvő sorszám szerint hajtja végre, függetlenül attól, hogy azokat milyen sorrendben gépeltük be. Ez alól idáig egy kivételt láttunk, amikor a GOTO kulcsszó után megadott sorszámnak megfelelő programsornál folytatódik a program végrehajtása.

Az 1.4 példaprogram tehát 3 programsorból és 6 utasításból áll.

Megjegyzés: A könyvben szereplő összes utasítás parancsként is adható, ezért erre külön nem hívjuk fel minden esetben a figyelmet.

Karakteres képernyőméret



A képernyőn 16 sorban (Ø., 1., 2., ..., 15. sor) soronként 64 (Ø., 1., 2., ..., 63. karakter), összesen 1Ø24 karakter jeleníthető meg. A megjeleníthető karaktereket a könyv végén levő táblázatban láthatjuk, ezekkel részletesebben a hetedik fejezetben foglalkozunk.

PRINT, INPUT

A **PRINT** és az **INPUT** utasítások segítségével a program futása közben is párbeszédet folytathatunk a számítógéppel.

Indítsuk el a következő programot!

1.6 példaprogram

```
10 CLS
20 PRINT"MELYIK PONT VILAGITSON ?"
30 PRINT"KEREM A KOORDINATAKAT !"
40 INPUT X
50 INPUT Y
60 CLS :
   SET(X,Y) :
   GOTO 20
```

Ez az 1.1 példaprogram továbbfejlesztett változata. A PRINT utasítással kérdéseket, feliratokat, karaktereket írathatunk ki a képernyőre. A kiírandó karaktereket idézőjelek közé kell tenni. A 4Ø-es sorban levő INPUT X utasítás hatására egy kérdőjel jelenik meg a képernyőn. Miután az X változónak adandó értéket begépeltük és elküldtük, kéri az Y változó értékét, és ha ezt is megadtuk, a program futása folytatódik. A 4Ø-es és az 5Ø-es sorokban levő utasításokat helyettesíthetjük a

4Ø INPUT X,Y

utasítással is. Ekkor a megjelenő kérdőjel után két, egymástól vesszővel elválasztott számot kell megadni. Amennyiben csak egy számot küldünk el, akkor a képernyőn két kérdőjel jelenik meg, és a számítógép várja a második szám megadását is. Ügyelni kell arra, hogy a begépelt számok a grafikus képernyőméreteknek megfelelő határok közé essenek. Ha ez teljesül, akkor a megfelelő pont kigyullad. A GOTO 2Ø utasítás eredményeképpen a számítógép annyiszor végzi el az eddigieket, ahányszor kívánjuk, és minden alkalommal másik pont gyújtható ki.

1.6 feladat: Írjunk olyan programot, amelyik megkérdezi négy pont koordinátáit, ezeket a pontokat kigyújtja, majd kezdi elölről mindezt!

1.7 feladat: Módosítsuk az 1.6 példaprogramot úgy, hogy amikor újabb pontot akarunk kigyújtani, akkor az előző pont továbbra is világítson! Mindig teljesíthető-e ez a feltétel?

Az utasítások begépelésekor a PRINT kulcsszó kérdőjellel (?) helyettesíthető, azonban a program listázásakor már a PRINT alak jelenik meg. Az üres (lista nélküli) PRINT utasítás a képernyőn egy soremelést eredményez. A PRINT utasítással a változók értékeit is ki lehet íratni, de ebben az esetben a változók nevét nem szabad idézőjelek közé tenni.

Futtassuk le és tanulmányozzuk a következő programot!

1.7 példaprogram

```
10 CLS
20 REM HAROM URES SOR KIIRATASA (SOREMELES)
30 PRINT : PRINT : PRINT
40 X=1 : Y=1
50 REM AZ X ES AZ Y VALTOZOK ERTEKEINEK KIIRATASA
60 PRINT"X=";X,"Y=";Y
70 GOTO 70
```

Láthatjuk tehát, hogy egy PRINT utasítás listája több kiírandó elemet is tartalmazhat. Ezeket vesszővel vagy pontosvesszővel kell elválasztani egymástól. A *pontosvessző hatására* az X, illetve az Y változók tartalma közvetlenül az egyenlőségjel után jelenik meg. A *vessző hatásáról* a következőt kell tudni:

A képernyő minden sora négy ún. nyomtatási zónára van osztva. Ezek a Ø., 16., 32. és a 48. karakterpozíción kezdődnek. Az 1.7 példaprogram futtatásakor az "X=" felirat és az X változó értéke az első zónába, az "Y=" felirat és az Y változó értéke pedig egy újabb zónába kerül. Ha a PRINT utasítás listájának végére pontosvesszőt írunk, akkor a következő felirat az előző kiírás után, ha vesszőt, akkor egy újabb nyomtatási zónában, ha pedig a lista végére egyik írásjel sem kerül, akkor egy újabb sorban folytatódik.



Az 1.7 példaprogramban már használtuk az áttekinthetőséget megnövelő **REM** utasítást. Egy programsornak a REM kulcsszó utáni részeit a gép nem hajtja végre, így oda tetszőleges magyarázó szöveget, megjegyzést írhatunk.

A képernyőn kérdéseket az INPUT utasítás segítségével is elhelyezhetünk, amit az INPUT kulcsszó után idézőjelek közé kell tenni:

1.8 példaprogram

```
10 CLS
20 INPUT"HANY CM MAGAS VAGY";C
30 CLS
40 PRINT C;"  CM MAGAS VAGY !"
```

1.8 feladat: Kati és Feri a következő játékot játsszák: A képernyőn megjelenő kérdésre Feri elfordul, és Kati két számot üt be. Ezután a képernyőn minden törlődik, és a megadott koordinátákkal rendelkező pont kigyullad. Feri visszafordul, és a képernyő bal felső sarkában kiíródó kérdésre: "MELYIK PONT VILAGIT?", megpróbálja eltalálni a világító pont koordinátáit. Megad két számot, majd ezután a számítógép képernyőt töröl, kiírja Kati számait és Feri számait. Készítsük el az ennek megfelelő programot!

1.9 feladat: Hol a hiba?

```
10 CLS
20 SET(0,0) : 2=X : 3=Y
30 SET(X,11) ; SET(X,21) , X=X+2
40 GOTO 30
```

1.10 feladat: Hol a hiba?

```
10 CLS
20 SET  (5,6) : S ET(5,7)
30 REZET(5,7) : GOTO 20
```

1.11 feladat: Hol a hiba?

```
10 CLS
20 INPUT X;Y
30 CLS : SET(X,Y)
40 GOTO 40
```

1.12 feladat: Hol a hiba?

```
10 CLS
20 PRINT
30 PRINT KERESD A HIBAT !
40 GOTO 50
```

1.13 feladat: Hol a hiba?

```
10 INPUT" A KOORDINATAK ?",X,Y
20 CLS : SET(X,Y)
30 GOTO 30
```

Gyakorlatok az első fejezet anyagához

1.1 gyakorlat: Készítsünk olyan programot, amelyik egy hatszög csúcsaiban helyez el pontokat és minden második pontot villogtat!

1.2 gyakorlat: Rajzoltassunk ki a képernyőre egy háromszöget!

1.3 gyakorlat: Készítsünk olyan programot, amelyik bekéri egy pont koordinátáit, majd innen kezdve kirajzol egy háromszöget!

1.4 gyakorlat: Készítsünk olyan programot, amelyik a képernyőn a lehető legtöbb pontot kigyújtja, de SET utasításból csak egyet használhatunk!

1.5 gyakorlat: Készítsünk olyan programot, amelyik kigyújt 15 darab pontot a képernyőn, majd fordított sorrendben kioltja azokat, és elölről kezdi mindezt! Találjunk ki minél érdekesebb elrendezéseket!

1.6 gyakorlat: Írassunk ki a képernyőre egy tetszőleges betűt grafikus pontokból!

1.7 gyakorlat: Írjunk olyan programot, amelyik bekéri két pont koordinátáit, majd ezeket felváltva villogtatja! Próbáljuk elérni, hogy a pontok kigyújtása, illetve lekapcsolása minél jobban érzékelhető legyen!

1.8 gyakorlat: Készítsünk olyan programot, amelyik egyszerre két függőleges és két vízszintes egyenes mindegyike mentén mozgat egy-egy pontot!

1.9 gyakorlat: Mozgassunk egy-egy pontot a képernyőn átlós irányban villogtatva, illetve folyamatosan!

1.10 gyakorlat: Készítsünk olyan programot, amelyik beolvassa egy pont koordinátáit, és ebből a pontból vízszintesen és függőlegesen pontokat mozgat!

1.11 gyakorlat: Mozgassuk egy négyzet négy csúcsát átlós irányban a képernyőn!

1.12 gyakorlat: Mozgassunk két pontot egymással szemben a képernyő két szélétől!

1.13 gyakorlat: Fusson nyolc pont a képernyő közepétől a négy sarka és az oldalak felezőpontja felé!

1.14 gyakorlat: Írassuk ki a képernyő közepére azt, hogy "SZIA!"

1.15 gyakorlat: Helyezzük el a képernyő közepén a "VILLOG" feliratot, és villogtassuk is azt!

1.16 gyakorlat: Írassuk ki a képernyőre a lehető legtöbb csillagot!

1.17 gyakorlat: Írassunk ki a képernyőre "A" karakterekből álló nagy A betűt!

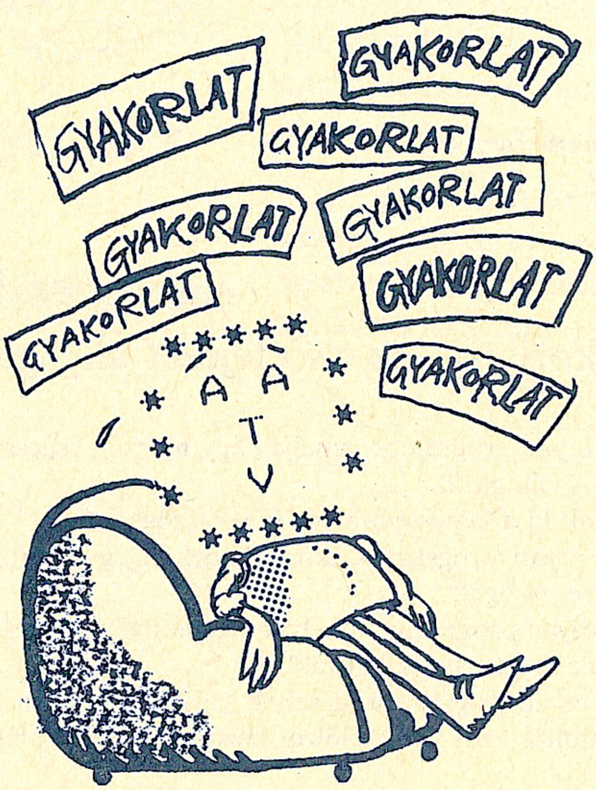
1.18 gyakorlat: A képernyő alján megjelenik egy vízszintes csík egyenlőségjelekből, majd ez felszalad, és kezdődik az egész elölről.
Készítsük el az ennek megfelelő programot!

1.19 gyakorlat: "/"-jelek mozogjanak "átlós" irányban a képernyőn!

1.20 gyakorlat: Olvastassuk be három változó értékét, és írassuk ki az alábbi formában:

a) A = 5
B = 6
C = 7

b) A = 5 B = 6 C = 7



Az első fejezet feladatainak egy lehetséges megoldása

1.1 feladat

```
10 CLS
20 SET(54,19) : SET(74,19) : SET(54,29) : SET(74,29) : SET(64,24)
30 GOTO 30
```

1.2 feladat

```
10 CLS
20 SET(64,24)
30 SET(54,19) : SET(74,19) : SET(54,29) : SET(74,29)
40 RESET(54,19) : RESET(74,19) : RESET(54,29) : RESET(74,29)
50 GOTO 30
```

1.3 feladat

```
10 CLS
20 SET(48,16) : SET(54,19) : SET(74,19) : SET(80,16) :
   SET(48,32) : SET(54,29) : SET(74,29) : SET(80,32) :
   SET(64,24)
30 GOTO 30
```

1.4 feladat

```
10 CLS
20 X=0 : Y=0
30 SET(X,23) : SET(50,Y) : RESET(X,23) : RESET(50,Y) :
   X=X+3 : Y=Y+1 :
   GOTO 30
```

1.5 feladat

```
10 CLS
20 X=1
30 SET(X,23) : RESET(X-1.23) :
   X=X+1 :
   GOTO 30
```

1.6 feladat

```
10 CLS
20 PRINT"KEREM A NEGY PONT KOORDINATAIT !"
30 INPUT X1,Y1 : INPUT X2,Y2 : INPUT X3,Y3 : INPUT X4,Y4
40 CLS
50 SET(X1,Y1) : SET(X2,Y2) : SET(X3,Y3) : SET(X4,Y4)
60 GOTO 20
```

1.7 feladat

```
10 CLS
20 PRINT"MELYIK PONT VILAGITSON ?"
30 PRINT"KEREM A KOORDINATAKAT !"
40 INPUT X,Y :
   SET(X,Y) :
   GOTO 40
```

Ha a már kigyújtott pontok között van olyan, amelyik abban a sorban van, amelyikben a következő INPUT utasítás kérdőjele megjelenik, akkor ez kialszik.

1.8 feladat

```
10 CLS
20 PRINT"    F E R I , FORDULJ EL !" : PRINT
30 PRINT"KEREM K A T I SZAMAIT !" : INPUT X,Y
40 CLS
50 PRINT"MELYIK PONT VILAGIT ?" :
   SET(X,Y) :
   INPUT A,B
60 CLS
70 PRINT"K A T I PONTJA :";X;Y : PRINT"F E R I PONTJA :";A;B
80 SET(X,Y) : SET(A,B)
90 GOTO 90
```

1.9 feladat

A 2Ø-as sorban az értékadó utasítások helyesen:

X=2 : Y=3

Az utasításokat a 3Ø-as sorban kettősponttal kell egymástól elválasztani.

1.10 feladat

A SET kulcsszó betűi között nem szabad üres helyet hagyni.

A 3Ø-as sorban a hibás REZET helyett a RESET kulcsszót kell írni.

1.11 feladat

A 2Ø-as sor INPUT utasításában a listaelemeket vesszővel kell egymástól elválasztani.

1.12 feladat

A PRINT utasítással kiírandó karaktereket idézőjelek közé kell tenni.

A GOTO utasítás nem létező sorszámra mutat.

1.13 feladat

Az INPUT utasítás végrehajtásakor mindenképpen megjelenik egy kérdőjel.

Az INPUT utasítást követő idézőjelek közé tett kérdés után mindig pontosvesszőt kell írni.

Második fejezet

HOGYAN MARADJUNK A KÉPERNYŐN?!

IF-THEN-ELSE

Módosítsuk az 1.6 példaprogramot úgy, hogy segítségével a képernyőnek csak az egyik felén lehessen pontokat kigyújtani (például azokat, amelyek X koordinátájára az X<64 feltétel teljesül)!



A módosított program alakja:

2.1 példaprogram

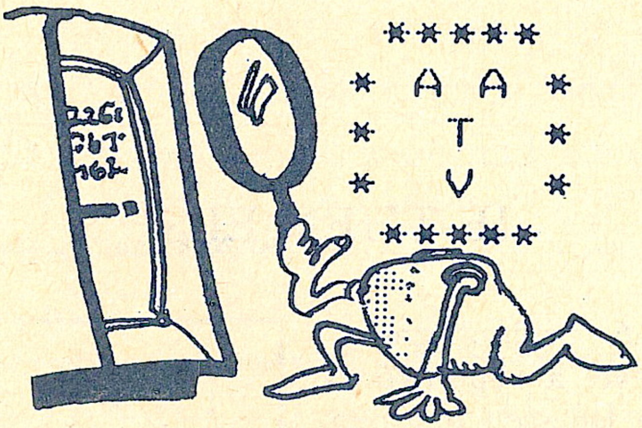
```
10 CLS
20 PRINT"MELYIK PONT VILAGITSON ?"
30 PRINT"KEREM A KOORDINATAKAT !"
40 INPUT X
50 IF X>63 THEN PRINT"X NAGYOBB 63 !" : GOTO 20 ELSE INPUT Y
60 CLS :
   SET(X,Y) :
   GOTO 20
```

Az 1.6 példaprogramnak az 5Ø-es sorát módosítottuk az ún. feltételes utasítás alkalmazásával.

Ha (IF) ismerjük az ebben szereplő kulcsszavak jelentését, *akkor* (THEN) máris könnyebb az utasítás értelmezése, *különben* (ELSE) pedig tanulmányozzuk figyelmesen ezt a mondatot! Ennek megfelelően, ha az 5Ø-es sorban az X > 63 feltétel teljesül, akkor a THEN kulcsszó utáni két utasítás hajtódik végre.

Ha pedig a feltétel nem teljesül, akkor beolvashatjuk a kigyújtandó pont másik koordinátáját is, és ezután a program végrehajtása a 60-as sorral folytatódik.

Vizsgáljuk meg a most megismert feltételes utasítás működését a következő két példaprogramban is!



2.2 példaprogram

```
10 CLS
20 PRINT"HANY EVES VAGY ?" : INPUT E
30 IF E>18 THEN PRINT"TEHAT MAR ELMULTAL 18 EVES !"
           ELSE PRINT"VAGYIS 19 EVESNEL FIATALABB VAGY !"
40 PRINT : PRINT"SOK SIKERT A KONYV TOVABBI OLVASASAHOZ !"
```

2.3 példaprogram

```
10 CLS
20 PRINT"AZ ELSO KOORDINATA "; : INPUT X
30 PRINT"A MASODIK KOORDINATA "; : INPUT Y
40 IF Y<20 THEN GOTO 30
           ELSE SET(X,Y) : GOTO 20
50 PRINT"MELYIK PONT VILAGIT ?"
```

2.1 feladat: A 2.3 példaprogram futtatásakor milyen esetben jelenik meg a "MELYIK PONT VILAGIT?" felirat?

Az első fejezet több programját kiegészíthetjük feltételes utasítással.

Az 1.4 példaprogram egy pontot mozgat. Ha az ottani 30-as sorból a RESET utasítást elhagyjuk, akkor a program egy vízszintes vonalat rajzol. Hasonló módon rajzol a következő példaprogramunk egy vonalat.

2.4 példaprogram

```
10 CLS
20 X=0
30 SET(X,42)
40 X=X+1
50 REM A KOORDINATA VIZSGALATA
60 IF X<127 THEN GOTO 30 ELSE PRINT"A VONAL HUZASAT BEFEJEZTEM !"
```

A program lefuttatása után örömmel tapasztalhatjuk, hogy a 6Ø-as sorban levő utasítással sikerült elkerülni azt, hogy futása hibajelzéssel fejeződjön be. Az ebben a példaprogramban is használt *feltételes utasításnak az általános alakja* a következő:

IF feltétel THEN 1.utasításcsoport ELSE 2.utasításcsoport.

Amennyiben a feltétel teljesül (IGAZ), azaz a 42. sorban még nem hagytuk el a 127. pontot, akkor a 3Ø-as sorra térve újabb pont kigyújtására van lehetőség (1.utasításcsoport), ha pedig a feltétel nem teljesül (HAMIS), akkor a program kiír egy feliratot a képernyőre (2.utasításcsoport).

Ha valamelyik utasításcsoport a

GOTO sorszám

utasításból áll, akkor a GOTO kulcsszó el is hagyható. Például a 2.4 példaprogram 6Ø-as sorában a THEN GOTO 3Ø helyett THEN 3Ø is írható.

2.2 feladat: Készítsünk olyan programot, amelyik 12 darab egymástól egyenlő távol levő szakaszt rajzol meg!

A most megismert utasítás feltétel része egy állítás, amelyben szerepelhetnek például a különböző relációjelek (<, >, =). Ügyeljünk arra, hogy itt az egyenlőségjelnek más értelme van, mint az értékadó utasításban!

A feltételes utasításban az

ELSE 2.utasításcsoport

rész el is maradhat. Ekkor, ha a feltétel HAMIS, a program a feltételes utasítást követő sorral folytatódik. Ennek megfelelően a 2.4 példaprogramot a következőképpen lehet módosítani:

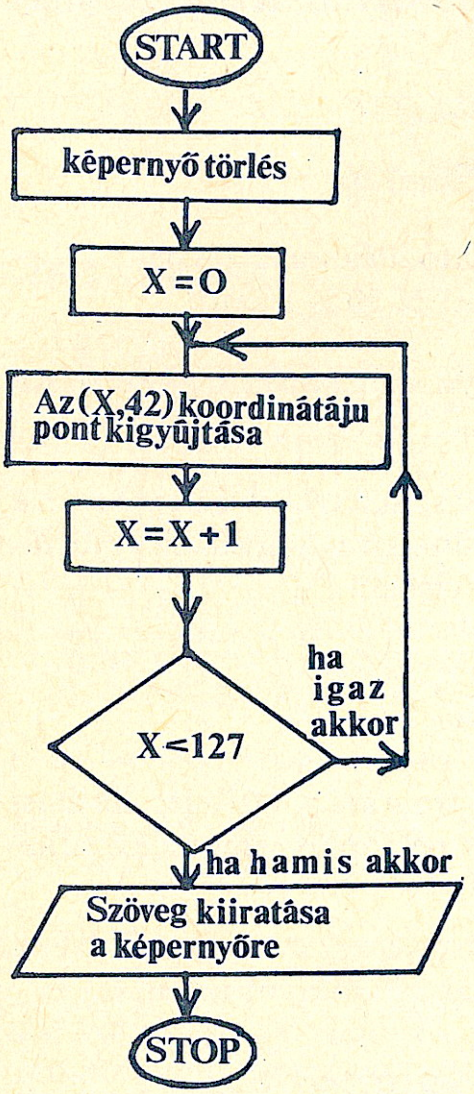
2.5 példaprogram

```
10 CLS
20 X=0
30 IF X<127 THEN SET(X,42) : X=X+1 : GOTO 30
40 PRINT"A VONAL HUZASAT BEFEJEZTEM !"
```

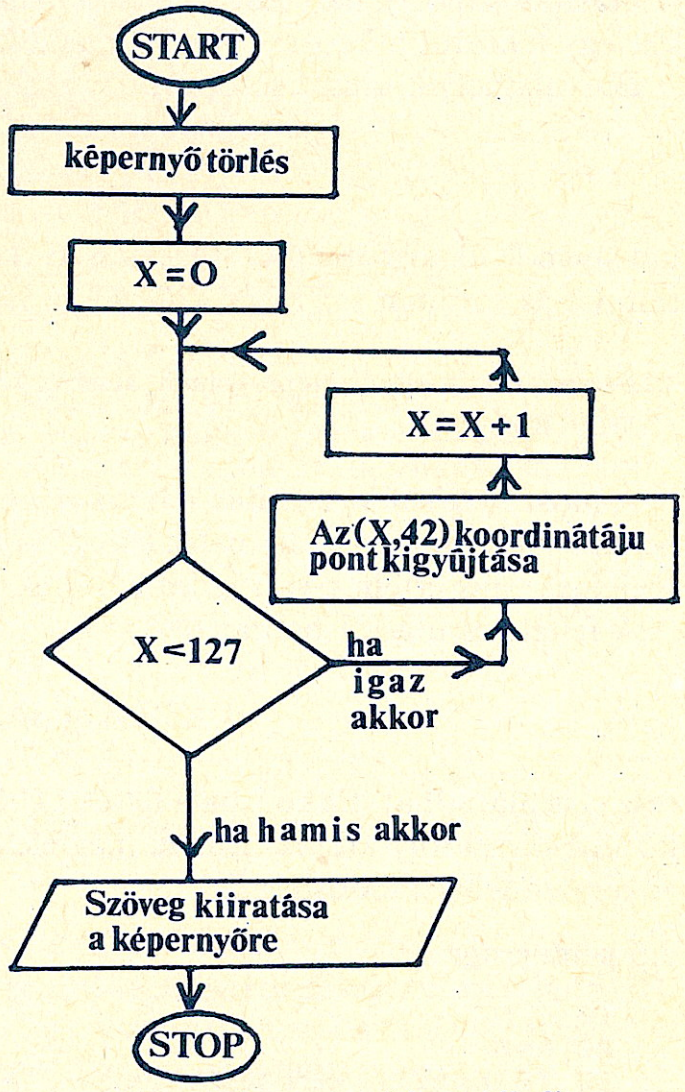
2.3 feladat: Módosítsuk az 1.8 feladatot, Kati és Feri játékát úgy, hogy a számítógép döntse el azt, hogy Feri eltalálta-e a világító pont koordinátáit! Ha nem, akkor újból próbálkozhat, ha igen, akkor ő helyezhet el egy pontot, s ezt Katinak kell kitalálnia.

Érdemes összehasonlítani a 2.4 és 2.5 példaprogramok folyamatábráját!

Látható, hogy mindkét folyamatábra ugyanazokból az utasításokból áll. Ezek között vannak olyanok, amelyeket egy futás során csak egyszer, másokat többször is végrehajt a számítógép. Mindkét program képernyőtörléssel és az X=Ø kezdőérték beállításával kezdődik. A 2.4 példaprogram ezután kigyújtja



A 2.5 példaprogram folyamatábrája



A 2.4 példaprogram folyamatábrája

a 42. sor Ø. pontját, majd eggyel megnöveli az X változó értékét és megvizsgálja, hogy az 127-nél kisebb-e. Ha igen, akkor kigyújtja a 42. sorban a következő pontot, ismét eggyel növeli az X változó értékét, és következik a vizsgálat. Mindaddig ismétli ezt, amíg a 127. pont nem kerülne kigyújtásra. Ekkor a vonal húzása befejeződik, amit egy felirat megjelenése is jelez a képernyőn, és a

program leáll. A 2.5 példaprogram látszólag ugyanezt végzi, de azzal a különbséggel, hogy ebben a vizsgálat, illetve a pont kigyújtása és a koordinátaérték növelésének sorrendje fel van cserélve. A két megoldás közötti különbség jól érzékelhető, ha az X=Ø helyett az X=127 utasítást írjuk. Ekkor a 2.4 példaprogram a (127,42) koordinátájú pontot kigyújtja, míg a 2.5 példaprogram egyetlen egyet sem gyújt ki.

A bemutatott folyamatábrákban a CLS és a PRINT utasítások közötti részt *ciklus*nak nevezzük. Az utasítások ilyen csoportjának általában az a jellemzője, hogy

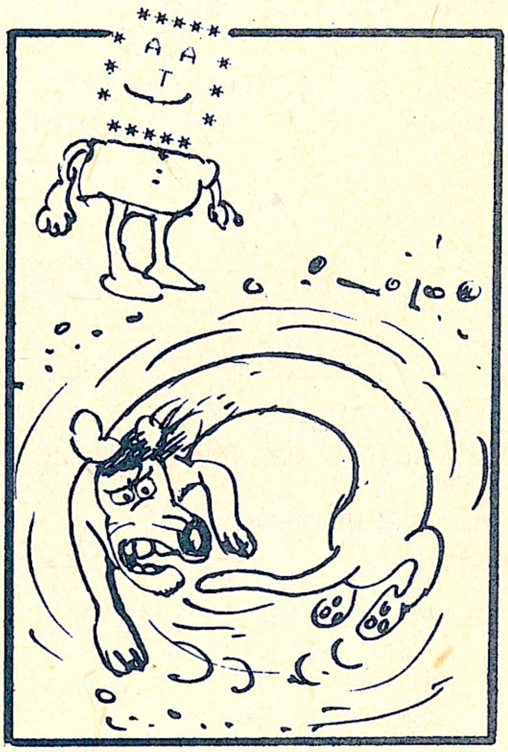
- egy változó, az ún. ciklusváltozó értékét a ciklus elején beállítjuk, ez a ciklusváltozó kezdőértéke;
- ennek értékét a cikluson belül adott értékkel (lépésköz) növeljük vagy csökkentjük mindaddig, amíg egy adott értéket (a ciklusváltozó végértéke) el nem ér;
- minden lépésben a cikluson belül további utasítások is végrehajtódhatnak.

A ciklusszervezéssel tehát lehetőség van arra, hogy utasításokat annyiszor hajtassunk végre, ahányszor azt kívánjuk, anélkül, hogy annyiszor le is írnánk azokat.

Az eddigi programokban már találkozhattunk olyan programrészekkel, amelyek a ciklusok különböző fajtáiból mutattak be néhányat. Már az első fejezet elején, az 1.1 példaprogramban szerepel a

30 GOTO 30

programsor. Tulajdonképpen ez is egy ciklus, méghozzá egy ún. végtelen ciklus. (Ott azt is elmondtuk, hogy ezt a [BREAK] billentyű vagy a [RESET] gomb segítségével szakíthatjuk meg. A végtelen ciklus futását a [SHIFT] és a [@] billentyűk egyidejű lenyomása is megszakítja.) Az említett utasítást az 1.1 példaprogramban azért használtuk, hogy a képernyőábránk szép legyen. Nem megfelelő, hibás programszervezés esetén azonban a végtelen ciklusok sok bonyodalmat is okozhatnak!



A 2.3 példaprogramban is a ciklusszervezés egy fajtája szerepel. A 20–40-es sorokban levő utasításokat addig ismétli a számítógép, amíg új koordinátákat adunk meg. Ezt a végtelen ciklust az előzőkhöz hasonlóan szakíthatjuk meg. Ez a program akkor is leáll, ha nem megfelelő adatokat adunk meg, és ekkor hibajelzést is kapunk.

STOP, END, CONT

A számítógépen a [BREAK] billentyű lenyomásával szakíthatjuk meg a program futását. Ekkor a képernyőn megjelenik a

Break in sorszám

üzenet annak a sornak a sorszámával, amelynek végrehajtása közben nyomtuk le a [BREAK] billentyűt.

A program futása leáll akkor is, ha egy **STOP** vagy egy **END** utasításhoz ér. Az első esetben a

Break in sorszám
READY

üzenet, a másodikban pedig csak a

READY

felirat jelenik meg a képernyőn. Ezekben az esetekben, ha a programot formailag nem módosítjuk, akkor a futásának folytatását a **CONT** [NL] paranccsal érhetjük el. Ekkor a program az utoljára végrehajtott utasítást követő utasítással folytatódik.

Észrevehettük, hogyha a programban sehol sem helyezünk el STOP vagy END utasítást, és nem tartalmaz végtelen ciklust, akkor a számítógép automatikusan úgy hajtja végre azt, mintha az utolsó sor után egy END utasítást tartalmazó sor lenne.

Megjegyzés: A STOP és az END utasítások hatása és szerepe egyes számítógéptípusoknál és programozási nyelvekben lényegesen különbözik!

FOR-TO-STEP-NEXT

Az 1.4, a 2.4 és a 2.5 példaprogramok vízszintesen rajzoltak egy pont mozgatásával vonalat. A következő példaprogramban egy új utasítás, a *ciklusutasítás* segítségével ezúttal egy ferde helyzetű vonalat rajzoltathatunk a képernyőre.

2.6 példaprogram

```
10 CLS
20 FOR X=4 TO 42 STEP 2 : REM A CIKLUS FEJE
30   SET(X+X+X,X)       : REM A CIKLUS MAGJA
40 NEXT X               : REM EZ IS A CIKLUS FEJE
50 GOTO 50
```

A 2Ø -as és a 4Ø -es sor, illetve a köztük levő 3Ø -as sor jelenti a ciklusutasítást. Az első kettő a *ciklus feje,* a köztük levő a *ciklus magja.* A ciklusmag utasításai mindaddig végrehajtódnak, amíg a ciklusváltozó (X) kezdeti értéke a megfelelő növelésekkel (lépésköz: 2) el nem éri a végső értékét (42). A lépésköz lehet pozitív és negatív érték is. Amennyiben értéke éppen egy, akkor el is hagyható. A negatív lépésköz például arra is felhasználható, hogy az átlót alulról felfelé rajzoltassuk meg. Próbálkozzunk meg vele!

2.4 feladat: Oldjuk meg a 2.2 feladatot ciklusutasítás segítségével is!

Láthatjuk tehát, hogy a ciklusutasítás alkalmazásával az előzőkben megismert ciklusszervezés egyszerűbbé válik. A ciklusutasítás elvégzi a ciklusváltozó módosítását és a vizsgálatot is. Jegyezzük azonban meg, hogy a ciklusutasításnál a ciklus magja legalább egyszer mindig végrehajtódik, és csak utána következik a vizsgálat. (Ez a 2.4 példaprogramban látott ciklusszervezésnek felel meg.)

Fontos, hogy jól megértsük a következő program működését!

2.7 példaprogram

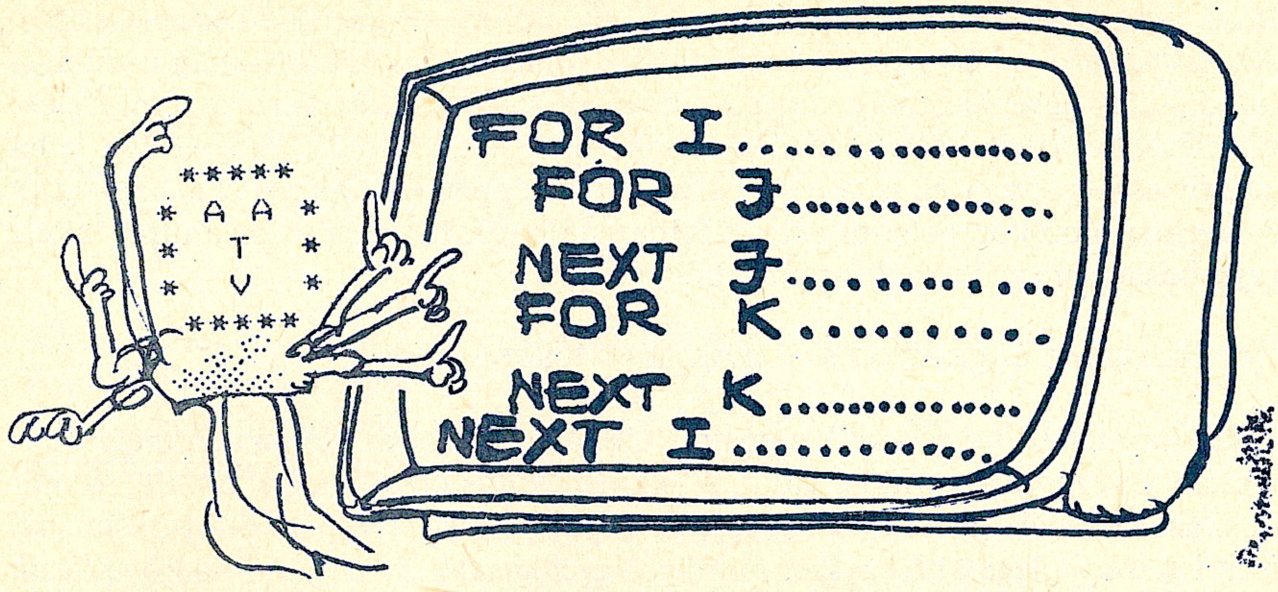
```
10 CLS
20 INPUT"KEZDOERTEK ";K
30 INPUT"VEGERTEK ";V
40 INPUT"LEPESKOZ ";L
50 FOR I=K TO V STEP L
60   PRINT I;
70 NEXT I
80 PRINT : PRINT"   VEGE"
```

Futtassuk le ezt a programot a következő értékekkel is!

K = 1,	V = 13,	L = 5;
K = 1,	V = − 13,	L = 5;
K = 1Ø,	V = 1,	L = 1!

A ciklus magja tartalmazhat egy újabb ciklust is. Ebben az esetben beszélünk *külső* és *belső ciklus*ról, a ciklusok ilyen elrendezését pedig egymásba skatulyázásnak nevezzük. Az *egymásba skatulyázott ciklus*okkal sok feladat egyszerűbben és áttekinthetőbben oldható meg.

A ciklusváltozó értékét a cikluson belül megváltoztatni nem szabad! Ebből következik, hogy az egymásba skatulyázott ciklusok ciklusváltozóinak különbözőeknek kell lenniük. Egy ciklusutasításba csak a FOR utasításon keresztül léphetünk be. Helyesen egymásba skatulyázott ciklusokra mutat példát a következő ábra:



2.5 feladat: Oldjuk meg a 2.2 feladatot egymásba skatulyázott ciklusok segítségével!

Programjaink futtatásánál a képernyőn egymás után megjelenő információk áttekinthetőségét, egymástól való jobb elkülöníthetőségét jelentősen megnövelhetjük például egy ciklusmag nélküli ún. *üres ciklusutasítás*sal is, amelynek segítségével egy pontot vagy egy feliratot adott ideig jeleníthetünk meg a képernyőn, vagy késleltethetjük a megjelenésüket.

2.6 feladat: Készítsünk programot, amelyik egy pontot villogtat 3 másodperces időközökkel!

Rövid összegzés

Próbáljuk megfejteni, hogyan működik a következő program. Ha sikerült, futtassuk le, és ellenőrizzük, hogy helyesen gondoltuk-e!

2.8 példaprogram

```
10 CLS
20 PRINT"MELYIK SORBA RAJZOLJAK VONALAT ?" : INPUT Y
30 IF Y<=-1 THEN 50
40 IF Y<48 THEN 60
50 PRINT"IRGUM-BURGUM ! JO ADATOKAT KEREK !!!" :
   GOTO 20
60 CLS
70 FOR X=0 TO 127
80   SET(X,Y)
90 NEXT X
100 FOR X=127 TO 0 STEP -1
110   RESET(X,Y)
120 NEXT X
130 PRINT"HOGY TETSZETT ?"
140 STOP
```

Itt felismerhettük csaknem az összes eddig tanult utasítást. Észrevehettük, ezt a programot nem lehet becsapni úgy, hogy a lehetségesnél nagyobb vagy kisebb koordinátákat adunk meg. (Lásd 3Ø –5Ø -es sorok!)

2.7 feladat: Készítsünk olyan programot, amelyik egy függőleges szakaszt balról jobbra mozgat! Az egymást követő helyzetekben a szakasz rajzolása felváltva alulról, illetve felülről történjen.

2.8 feladat: Írjunk olyan programot, amelyik 2Ø darab vízszintes szakaszt rajzol! Az egymás után következő szakaszok növekvő hosszúságúak legyenek, és a növekmény legyen állandó.

2.9 feladat: Készítsünk olyan programot, amelyik egy átlós helyzetű szakaszt mozgat a képernyőn!

2.10 feladat: Hol a hiba?

```
10 CLS : Y=0
20 SET(60,Y)
30 IF Y<47 THEN Y=Y+1 : GOTO 20 ELSE 40
40 GOTO 40
```

2.11 feladat: Hol a hiba?

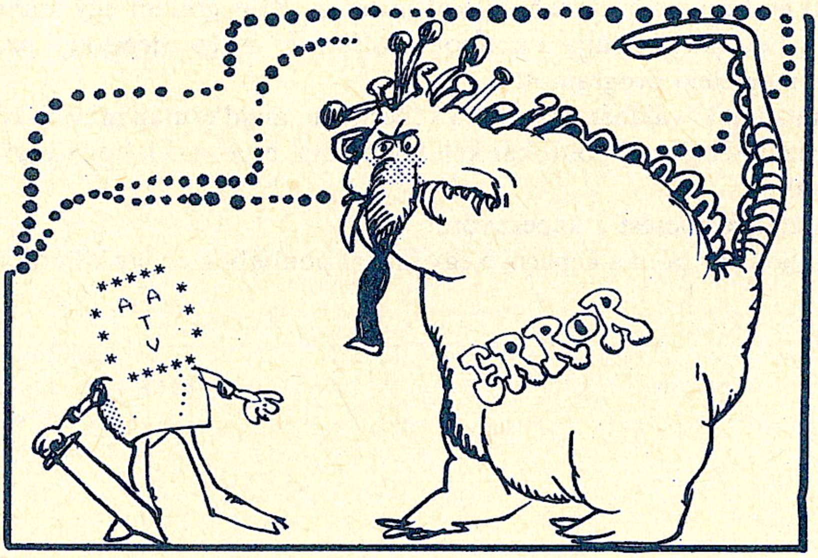
```
10 INPUT X
20 IF X>10 THEN GOTO 40 : PRINT "X NAGYOBB TIZNEL"
30 PRINT "X NEM NAGYOBB TIZNEL"
40 FOR K=1000 TO 0 STEP -2 : NEXT K
50 GOTO 10
```

2.12 feladat: Hol a hiba?

```
10 CLS
20 FOR X=2 TO 42 STEP 2
30   SET(X+X,X)
40 NEXT
50 GOTO 50
```

2.13 feladat: Hol a hiba?

```
10 CLS
20 FOR Y=48 TO 0 STEP -1
30   FOR X=128 TO 0 STEP -1
40     SET(X,Y)
50 NEXT X,Y
60 GOTO 40
```



Gyakorlatok a második fejezet anyagához

2.1 gyakorlat: Készítsünk olyan programot, amelyik kirajzol a képernyőre 5 függőleges vonalat, majd törli a képernyőt, és kirajzol 5 vízszintes vonalat! Ezután ezt ismételgeti állandóan.

2.2 gyakorlat: Rajzoltassunk 15 olyan vonalat, amelyek 36 részre bontják a képernyőt!

2.3 gyakorlat: Írjunk olyan programot, amelyik 1Ø darab vízszintes vonalat rajzol, de úgy, hogy mindegyik elé egy számot is ír!

2.4 gyakorlat: Írjunk olyan programot, amelyik a képernyő négy sarka és az oldalak felezőpontja felé vonalakat húz a képernyő középpontjából!

2.5 gyakorlat: Húzassunk vízszintes vonalakat egymás alá felváltva mínusz jelekből (−), illetve grafikus pontokból!

2.6 gyakorlat: Rajzoltassunk keretet a képernyőre grafikus pontokból!

2.7 gyakorlat: Rajzoltassunk keretet a képernyőre csillag karakterekből (*)!

2.8 gyakorlat: Írassuk ki a képernyőre a megfelelő sorba a következő feliratokat: "ELSO SOR", "HARMADIK SOR", "OTODIK SOR", majd grafikus pontokkal húzassuk is alá őket!

2.9 gyakorlat: Készítsünk programot, amelyik koordinátatengelyeket rajzol!

2.10 gyakorlat: Egy program olvassa be egy pont koordinátáit és ábrázolja azt egy koordináta-rendszerben!

2.11 gyakorlat: Írassuk ki a képernyőre a "MASSALHANGZOK" szót, majd grafikus pontokkal takarjuk le a magánhangzóit!

2.12 gyakorlat: Írassunk ki három számot a képernyőre, és mindegyiket keretezzük be csillag karakterekkel (*)!

2.13 gyakorlat: Írassunk ki a képernyő első sorába 5 darab pluszjelet (+), ezután ezek 3 másodpercenként mindig egy sorral lejjebb jelenjenek meg!

2.14 gyakorlat: A képernyőn megjelenik az 1-es szám, majd három másodpercenként mindig az előzőnél eggyel nagyobb szám. Készítsük el ezt a programot!

2.15 gyakorlat: Készítsünk programot, amelyik metronómként működik!

2.16 gyakorlat: Készítsünk programot, amelyik óraként működik!

2.17 gyakorlat: Írjunk olyan programot, amelyik bekéri a pontos időt, majd mutatja az idő múlását!

2.18 gyakorlat: 5 másodpercenként szaladjon fel egy vízszintes vonal a képernyőn!

2.19 gyakorlat: Írjunk olyan programot, amelyik 3 vízszintes vonalat rajzol egyszerre úgy, hogy a k. másodpercben $2k$ darab újabb grafikus pontot gyújt ki vonalanként!

2.20 gyakorlat: Írassunk ki a képernyőre 5Ø darab csillagot, de csak egy PRINT utasítást, és abban is csak egy csillag karaktert használhatunk!

2.21 gyakorlat: Kati és Feri a következő játékot játsszák: Kati gondol egy számot, és azt közli a számítógéppel. Feri megpróbálja a számot kitalálni, és a gép ellenőrzi a próbálgatásait. Készítsük el a megfelelő programot!

2.22 gyakorlat: Gyújtsunk ki valahány pontot a képernyőn, majd ezután mindig kétszer annyit, mint az előző lépésben! Mindig új pontokat kell kigyújtani, és az a cél, hogy végül a lehető legtöbb pont világítson!

2.23 gyakorlat: Imitáljunk hóesést a képernyőre!

2.24 gyakorlat: Induljon egy pont a képernyő egy kívánt pontjából, és egy biliárdgolyó nyomvonalát rajzolja fel!

A második fejezet feladatainak egy lehetséges megoldása

2.1 feladat

Ha a programot a RUN [NL] paranccsal futtatjuk, akkor az 5Ø-es sorra sohasem kerül a végrehajtás. Lehetőség van azonban arra, hogy a RUN5Ø [NL] vagy a GOTO5Ø [NL] parancsok valamelyikével indítsuk a program futását. Ekkor a képernyőn rögtön megjelenik a kívánt felirat. (Természetesen akkor is megjelenik, ha kilistázzuk a programot.)

2.2 feladat

```
10 CLS
20 X=25 : Y=6
30 SET(X,Y)
40 X=X+1
50 IF X=50 THEN 60 ELSE 30
60 X=25 : Y=Y+3
70 IF Y=42 THEN 80 ELSE 30
80 GOTO 80
```

2.3 feladat

```
10 CLS
20 J=1
30 PRINT"F E R I  , FORDULJ EL !" : PRINT :
   PRINT"KEREM K A T I SZAMAIT !"
40 INPUT X,Y
50 CLS
60 PRINT"MELYIK PONT VILAGIT ?"
70 SET(X,Y)
80 INPUT A,B
90 IF X=A THEN 100 ELSE 50
100 IF Y=B THEN PRINT"UGYES VAGY!" : I=0 ELSE 50
110 I=I+1 : IF I<300 THEN 110
120 IF J=2 THEN 10 ELSE J=2
130 CLS
140 PRINT"K A T I  , FORDULJ EL !" : PRINT :
    PRINT"KEREM F E R I SZAMAIT !"
150 INPUT X,Y
160 GOTO 50
```

2.4 feladat

```
10 CLS
20 X=25
30 FOR Y=6 TO 39 STEP 3
40   SET(X,Y)
50 NEXT Y
60 X=X+1
70 IF X<51 THEN 30
80 GOTO 80
```

2.5 feladat

```
10 CLS
20 FOR X=25 TO 50
30   FOR Y=6 TO 39 STEP 3
40     SET(X,Y)
50   NEXT Y
60 NEXT X
70 GOTO 70
```

2.6 feladat

```
10 CLS
20 SET(24,24)
30 FOR I=1 TO 1000 : NEXT I
40 RESET(24,24)
50 FOR I=1 TO 1000 : NEXT I
60 GOTO 20
```

A villogtatáshoz szükséges idővároshoz felhasználható a 2.3 feladat megoldásánál már látott, feltételes utasítást használó módszer is.

2.7 feladat

```
10 CLS
20 I=1
30 FOR X=30 TO 90 STEP 5
40   IF I=2 THEN 130
50   FOR Y=25 TO 5 STEP -1
60     RESET(X-5,Y)
70   NEXT Y
80   FOR Y=5 TO 25
90     SET(X,Y)
100   NEXT Y
110   I=2
120   GOTO 200
130   FOR Y=5 TO 25
140     RESET(X-5,Y)
150   NEXT Y
160   FOR Y=25 TO 5 STEP -1
170     SET(X,Y)
180   NEXT Y
190   I=1
200 NEXT X
210 GOTO 210
```

2.8 feladat

```
10 CLS
20 I=10
30 FOR Y=3 TO 41 STEP 2
40   FOR X=9 TO I+9
50     SET(X,Y)
60   NEXT X
70   I=I+5
80 NEXT Y
90 GOTO 90
```

2.9 feladat

```
10 CLS
20 I=0
30 IF I=0 THEN 70
40 FOR X=0 TO 10
50   RESET(I-5+X+X,X+20)
60 NEXT X
70 FOR X=0 TO 10
80   SET(I+X+X,X+20)
90 NEXT X
100 I=I+5
110 IF I<105 THEN 30 ELSE 10
```

2.10 feladat

A 2Ø-as sorban nulla helyett *O*-betű van.

2.11 feladat

A 2Ø-as sorban logikai hiba van. A PRINT"X NAGYOBB TIZNEL" utasítás soha nem hajtódik végre.

2.12 feladat

Sehol!!! A NEXT után a ciklusváltozót nem kötelező kiírni, amennyiben egyértelmű, hogy ez a kulcsszó melyik ciklus fejéhez tartozik.

2.13 feladat

A 4Ø-es sorban az X és az Y változók értéke a megengedettnél nagyobb.
A 6Ø-as sorban levő GOTO4Ø utasítás a ciklusmagba ugrik. Ez tilos!

Megjegyzés: Egymásba skatulyázott ciklusok esetén az egymás után következő NEXT utasítások összevonhatók. Ebben az esetben először a legbelső ciklus, majd az azt tartalmazó ciklus, ... változója kerül az egyetlen NEXT kulcsszó után. Ilyenkor a ciklusváltozókat egymástól vesszővel kell elválasztani.

Harmadik fejezet

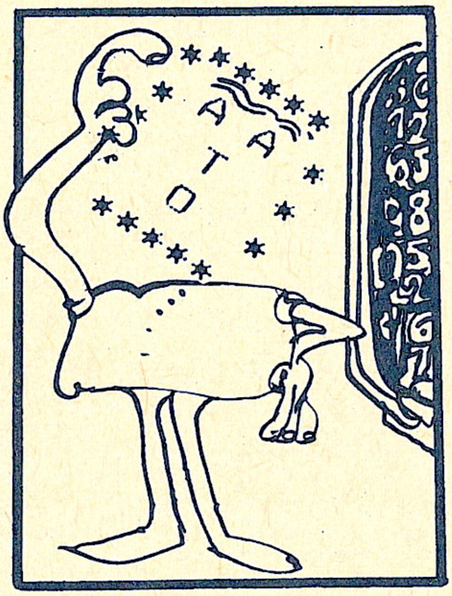
ALAKZATOK A KÉPERNYŐN

Rajzolási sebesség

Tanulmányozzuk át a következő programot!

3.1 példaprogram

```
10 CLS
20 X=20 : Y=5
30 SET(X,Y) :
   X=X+1
40 IF X=90 THEN X=20 : Y=Y+1 ELSE 30
50 IF Y=37 THEN 60 ELSE 30
60 GOTO 60
```



Ez a program egy téglalapot rajzol. Futtassuk le, és ellenőrizzük, hogy ezt kb. 40 másodperc alatt végzi el!

Ugye nem tetszett, hogy milyen lassan rajzolta meg az ábrát?! Az előző fejezetben megismertük a ciklusszervezés utasításait. Ezeknek alkalmazásával ismét rajzoltathatunk téglalapot a számítógéppel.

3.2 példaprogram

```
10 CLS
20 FOR Y=5 TO 36
30   FOR X=20 TO 89 : SET(X,Y) : NEXT X
40 NEXT Y
50 GOTO 50
```

Ha ezt a programot lefuttatjuk, örömmel tapasztalhatjuk, hogy az előzővel megegyező méretű téglalapot kb. fele annyi idő alatt sikerül kirajzoltatni (kb. 20 másodperc).

Megjegyzés: Az itt és a továbbiakban közölt időértékek csak tájékoztató jellegűek, különböző gépeknél néhány másodperces eltérések lehetségesek. A megadott időadatokat érdekes és érdemes összehasonlítani a könyv tanulmányozásakor használt számítógépen kapott értékekkel!

3.1 feladat: Készítsünk olyan programot, amelyik két satírozott téglalapot rajzol minél gyorsabban úgy, hogy ezek egy T alakot alkossanak!

DEFINT

Az eddigi programjainkban mindig olyan változókat használtunk, amelyek értéke lehetett egész vagy nem egész valós szám is. Ezeket a változókat *valós típusú változók*nak nevezzük. Egy programon belül azonban bizonyos változókat definiálhatunk *egész típusú változók*ként is. Ezeknek értéke mindig egész szám.

A 3.2 példaprogramot egészítsük ki az alábbi sorral:

5 DEFINT X,Y

A fenti **DEFINT** definiáló utasítás hatására a számítógép az X és az Y változókat – valamint az összes X-szel, illetve Y-nal kezdődő változót – egész típusúként kezeli. Az egész típusú változók ábrázolásához kevesebb helyre van szükség a memóriában, és az azokkal végzett műveletek végrehajtása gyorsabb, mint a valós típusú változókkal történőké. Akkor is, ha ezek tartalma csupa egész szám! Erről magunk is meggyőződhetünk, mert az előbbi 5-ös sorral kiegészített 3.2 példaprogram kb. 15 másodperc alatt rajzolja ki a téglalapot.

Egyelőre csak annyit árulunk el, hogy később majd megismerkedünk olyan utasításokkal is, amelyek segítségével tetszőleges téglalap kirajzolásához még egy másodpercre sem lesz szükség...

Ha egy programban több olyan változót akarunk egész típusúként definiálni, amelyek nevei az *ábécé*-ben egymás utáni betűkkel kezdődnek, akkor ezt megtehetjük anélkül is, hogy a DEFINT kulcsszó után felsorolnánk valamennyit.

Például a

DEFINT I,J,K,L,M

utasítás egyenértékű a

DEFINT I–M

utasítással.

3.2 feladat: Készítsünk (8 × 8-as) sakktáblát rajzoló programot! Törekedjünk arra, hogy a számítógép minél gyorsabban végezze el a rajzolást!

3.3 feladat: Rajzoltassunk a számítógéppel téglalapspirált!

Megjegyzés: Az egész és a valós típusú változók használatával részletesebben a következő fejezetekben találkozunk.

POINT

Ismerkedjünk meg a **POINT** függvénnyel! A POINT(X,Y) értéke −1, ha az (X,Y) koordinátájú pont világít, és Ø, ha nem. Ezek alapján fejtsük meg a következő program működését:

3.3 példaprogram

```
10 CLS
20 DEFINT X,Y
30 FOR Y=5 TO 36
40   FOR X=20 TO 89 : SET(X,Y) : NEXT X
50 NEXT Y
60 FOR Y=0 TO 47
70   FOR X=0 TO 127
80     IF POINT(X,Y)=-1 THEN RESET(X,Y) ELSE SET(X,Y)
90 NEXT X,Y
100 GOTO 100
```

Ha sikerült, ellenőrizzük a megfejtést: futtassuk le a programot!

3.4 feladat: Módosítsuk a 3.3 feladatra adott megoldást úgy, hogy miután megrajzoltattuk a téglalapspirált, a számítógép készítse el annak negatívját is!

AND és OR

A POINT(X,Y) függvényutasítás segítségével tehát arra kapunk választ, hogy az (X,Y) koordinátájú pont világít-e. A 3.3 példaprogram 8Ø-as sorát vizsgálva:

```
8Ø IF POINT(X,Y) = -1 THEN RESET(X,Y) ELSE SET(X,Y)
```

azt láthatjuk, hogyha a feltétel IGAZ, vagyis a pont világít, akkor eloltja azt, ha pedig HAMIS, vagyis a pont nem világít, akkor kigyújtja. A POINT(X,Y) egy logikai állítás is, amelynek két lehetséges értéke van, az IGAZ és a HAMIS. Ennek megfelelően a 8Ø-as sorral egyenértékű a következő sor:

```
8Ø IF POINT(X,Y) THEN RESET(X,Y) ELSE SET(X,Y)
```

mivel a feltételes utasítás feltétel része is egy logikai állítás. Az AND és az OR *logikai műveletek*kel logikai állításokat kapcsolhatunk össze, így az IF kulcsszó után összetett feltétel is szerepelhet. Ezek segítségével több feltételes utasítást eggyel helyettesíthetünk. Erre nézzünk most két példát:

Az

```
IF X>1Ø OR X<5 THEN 2Ø ELSE 9Ø
```

utasítás hatására a számítógép a programot a 2Ø-as sornál folytatja, ha az X változó értéke nagyobb 1Ø-nél vagy kisebb 5-nél, ha pedig az X változó értéke 5-nél nem kisebb és 1Ø-nél nem nagyobb, akkor a 9Ø-es sornál.

Az

```
IF X>15 AND X<3Ø THEN 2Ø ELSE 9Ø
```

utasítás hatására a program futása a 2Ø-as sorban folytatódik, ha az X változó értéke 15 és 3Ø közé esik, minden egyéb esetben a 9Ø-es sorra tér rá. Az AND és az OR (magyar megfelelőjük az ÉS, illetve a VAGY) kétváltozós logikai műveletek, könnyű megjegyezni az ún. igazságtáblázatukat:

AND	IGAZ	HAMIS
IGAZ	IGAZ	HAMIS
HAMIS	HAMIS	HAMIS

OR	IGAZ	HAMIS
IGAZ	IGAZ	IGAZ
HAMIS	IGAZ	HAMIS

Tehát egy *A* AND *B* típusú logikai állítás csakis akkor IGAZ, ha az *A* és a *B* állítás is IGAZ, egy *A* OR *B* típusú logikai állítás pedig akkor IGAZ, ha az *A* és a *B* állítások közül legalább az egyik IGAZ.

3.5 feladat: Rajzoltassunk a számítógéppel egy satírozott négyzetet, amelyik tartalmaz egy üres négyzetet! Az ábra kb. 75 másodperc alatt készüljön el. (Ügyeljünk arra, hogy a négyzetek a rajzolási lehetőségekhez képest valóban négyzetek legyenek!)

3.6 feladat: Oldjuk meg a 3.1 feladatot logikai műveleteket tartalmazó feltételes utasítások segítségével!

3.7 feladat: Hol a hiba?

```
10 CLS
20 X=1 : Y=2
30 IF X<120 THEN SET(X,Y) : X=X+1 : GOTO30
40 DEFINT X,Y
```

3.8 feladat: Hol a hiba?

```
10 CLS
20 DEFINT Z-A
30 X=1 : Y=1
40 IF X<55 AND Y<35 THEN SET(X,Y) : X=X+1 : Y=Y+1 : GOTO 40
50 FOR X=0 TO 127
60   FOR Y=47 TO 0 STEP -1
70     IF POINT X,Y THEN SET(X,Y) ELSE RESET(X,Y)
80 NEXTX,Y
90 GOTO 90
```

Gyakorlatok a harmadik fejezet anyagához

3.1 gyakorlat: Gyűjtsuk ki a képernyő összes grafikus pontját több különböző módon is!

3.2 gyakorlat: Satírozott téglalapok segítségével rajzoltassunk F és E betűt úgy, hogy az E betűt kétszer annyi ideig rajzolja a számítógép, mint az F betűt!

3.3 gyakorlat: Osszuk két vonallal négy részre a képernyőt, majd két, csúcsban érintkező részt satírozzunk be! Törekedjünk arra, hogy mindez minél gyorsabban történjen.

3.4 gyakorlat: 1000 darab felkiáltójel jelenjen meg minél gyorsabban a képernyőn!

3.5 gyakorlat: Rajzoltassunk minél gyorsabban állandó szélességű keretet a képernyőre!

3.6 gyakorlat: Mozogjon egy pont a képernyő két pontja között felváltva gyorsulva majd lassulva!

3.7 gyakorlat: Járja be egy pont egy téglalap kerületét, majd tegye ugyanezt ellenkező irányban és tizedannyi idő alatt! Ezután ezt ismételgesse!

3.8 gyakorlat: Rajzoltassunk téglalapspirált szaggatott vonal segítségével a képernyőre!

3.9 gyakorlat: Készítsünk programot, amelyik téglalapspirált rajzol a képernyőre de úgy, hogy először a vízszintes, majd ezután a függőleges vonalakat rajzolja meg!

3.10 gyakorlat: Írjunk olyan programot, amelyik olyan téglalapspirált rajzol, amelyben a másodiktól kezdve minden szakasz hossza az azt megelőző szakasz hosszának kétszerese!

3.11 gyakorlat: Rajzoltassunk egy házat a képernyőre!

3.12 gyakorlat: Rajzoltassunk a képernyőre minél több különböző nagyságú négyzetet! (A négyzetek nem érintkezhetnek egymással!)

3.13 gyakorlat: Jelenjen meg a képernyőn egy ablak, amin a redőnyt állandóan le- és felhúzzák!

3.14 gyakorlat: A képernyőn három téglalap közül kettő fogy és egy növekszik, majd egy fogy és kettő növekszik periodikusan úgy, hogy mindig ugyanannyi pont világít. Készítsük el ezt a programot!

3.15 gyakorlat: Rajzoltassunk működő homokórát a képernyőre!

3.16 gyakorlat: Rajzoltassunk egy négyzetet és egy egyenest a képernyőre, majd adott időközönként jelenjen meg a négyzetnek az egyenesre vonatkozó tükörképe!

3.17 gyakorlat: A képernyő közepén jelenjen meg négy darab négyzet, majd mozogjanak ezek a képernyő négy sarka felé!

3.18 gyakorlat: Jelenjen meg a képernyőn egy téglalap, majd csúsztassuk el annak rétegeit egymáson úgy, hogy egy paralelogrammához hasonló alakzat adódjon!

3.19 gyakorlat: Egy vízszintes és egy függőleges szakasz mozogjon egymásra merőlegesen a képernyőn!

3.20 gyakorlat: Rajzoltassunk a képernyőre 9 szakaszt úgy, hogy azok legalább 20 pontban messék egymást! A metszéspontok ne világítsanak.

3.21 gyakorlat: Rajzoltassunk a képernyőre 5 különböző alakzatot egymás után úgy, hogy bármelyik két alakzatnak legyen közös pontja! Ha egy kigyújtandó pont rajzolás közben már világít, kapcsoljuk ki!

3.22 gyakorlat: Gyújtsuk ki valamely sor első száz grafikus pontját, ezután kapcsoljuk ki ezen száz pont minden másodikját, általában a *k*. lépésben minden *k*. pontot vizsgáljunk meg a századikig bezárólag, s ha a pont világított, akkor kapcsoljuk ki, ha nem, akkor gyújtsuk ki! 100 ilyen lépés után írassuk is ki a világító pontok koordinátáit!

3.23 gyakorlat: Gyújtsuk ki a képernyő azon pontjait, amelyek X koordinátái 5-nél nagyobbak, de 15-nél kisebbek, vagy pedig 81 és 93 közé esnek, Y koordinátái pedig 6 és 14 közötti számok!

A harmadik fejezet feladatainak egy lehetséges megoldása

3.1 feladat

```
10 CLS
20 FOR Y=2 TO 5
30   FOR X=10 TO 107 : SET(X,Y) : NEXT X
40 NEXT Y
50 FOR Y=6 TO 42
60   FOR X=56 TO 65 : SET(X,Y) : NEXT X
70 NEXT Y
80 GOTO 80
```

3.2 feladat

```
10 CLS
20 DEFINT I-L
30 K=0
40 FOR J=1 TO 4
50   FOR I=1 TO 8
60     FOR L=I TO 48+I STEP 16
70       SET(31+L,6+K+J) : SET(39+L,10+K+J)
80 NEXT L,I,J
90 IF K<24 THEN K=K+8 : GOTO 40
100 GOTO 100
```

3.3 feladat

```
10 CLS
20 DEFINT X,Y
30 X1=0 : X2=47
40 FOR X=X1 TO X2 : SET(40+X,X1) : SET(40+X2,X) : NEXT X
50 X1=X1+2
60 FOR X=X2 TO X1 STEP -1 : SET(40+X,X2) : SET(40+X1,X) : NEXT X
70 X2=X2-2
80 IF X2>24 THEN 40
90 GOTO 90
```

3.4 feladat

```
10 CLS
20 DEFINT X,Y
30 X1=0 : X2=47
40 FOR X=X1 TO X2 : SET(40+X,X1) : SET(40+X2,X) : NEXT X
50 X1=X1+2
60 FOR X=X2 TO X1 STEP -1 : SET(40+X,X2) : SET(40+X1,X) : NEXT X
70 X2=X2-2
80 IF X2>24 THEN 40
90  FOR X=0 TO 127
100   FOR Y=0 TO 47
110     IF POINT(X,Y)=-1 THEN RESET(X,Y) ELSE SET(X,Y)
120 NEXT Y,X
130 GOTO 130
```

3.5 feladat

```
10 CLS
20 DEFINT X,Y
30 X=31 : Y=7
40 IF Y>10 AND Y<24 THEN 50 ELSE 60
50 IF X>40 AND X<62 THEN 70
60 SET(X,Y)
70 X=X+1
80 IF X<90 THEN 40 ELSE X=31 : Y=Y+1
90 IF Y<41 THEN 40
100 GOTO 100
```

3.6 feladat

```
10 CLS
20 X=10 : Y=2
30 IF X<56 AND Y>5 THEN 60
40 IF X>65 AND Y>5 THEN 60
50 SET(X,Y)
60 X=X+1
70 IF X<108 THEN 30 ELSE X=10 : Y=Y+1
80 IF Y<43 THEN 30
90 GOTO 90
```

3.7 feladat

Amennyiben a program futása során az X és az Y változókat egész típusúnak kívánjuk definiálni, úgy a DEFINT utasítást értelemszerűen a program elejére kell tenni.

3.8 feladat

A 20-as sorban az A és a Z betű sorrendje felcserélődött.

A POINT kulcsszó után az X és az Y változókat zárójelbe kell tenni.

A 70-es sorban a SET utasítást az ELSE-ágba, a RESET utasítást pedig a THEN-ágba kell írni ha azt akarjuk, hogy ez a sor a vizsgált pontot kigyújtsa amennyiben az nem világított, és kikapcsolja ha világított.

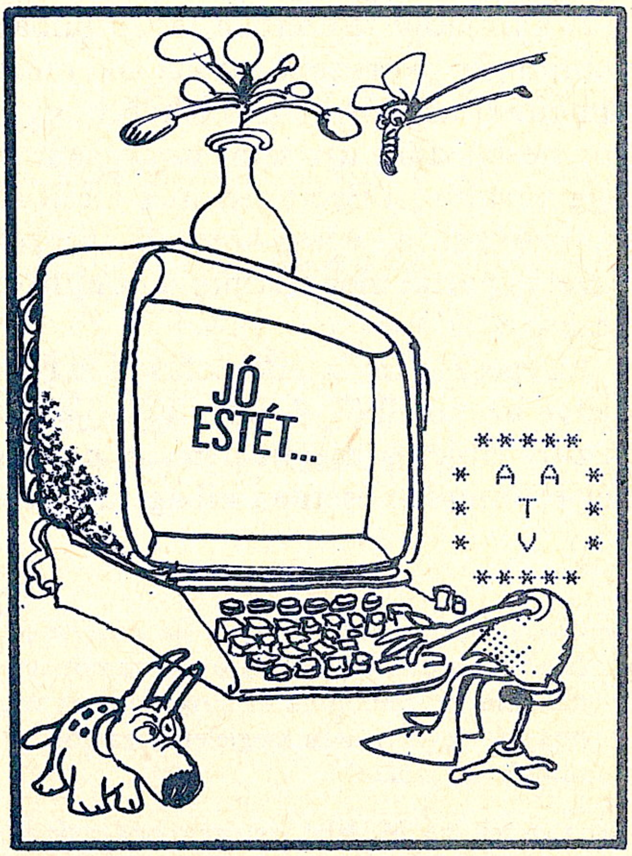
A 80-as sorban a NEXT utasítás után az X és az Y változók sorrendjét fel kell cserélni.

Negyedik fejezet

A SZÁMÍTÓGÉP SZÁMOLNI IS TUD

Az egész típusú változók

Eddig főleg olyan feladatokkal találkoztunk, amelyek segítségével a számítógép grafikus funkcióit ismerhettük meg. Ez nemcsak azért volt fontos, hogy a számítógép grafikájának alapjaival tisztában legyünk, hanem azért is, hogy jól tudjunk tájékozódni a képernyőn, ismerjük annak ábrázolási méreteit. Ne feledjük el, a számítógéppel folytatott párbeszéd során a gép képernyője az elsődleges információforrás a külvilág, így a mi számunkra is!



A DEFINT utasítással már az előző fejezetben megismerkedtünk, de egy változó egész típusú lesz akkor is, ha a program során végig százalékjelet (%) írunk a változó neve után. Ügyeljünk azonban arra, hogyha a DEFINT utasítással a D változót nem definiáltuk előzetesen egész típusúként, akkor a D és a D% változók különböző változókat jelentenek. Erről a következő parancsok segítségével akár magunk is meggyőződhetünk:

```
D=16.6         NL
D%=16.6        NL
PRINT D,D%     NL
```

Megjegyzések:

- Az utasításokhoz hasonlóan a parancsokat is írhatjuk egy sorba, egymástól kettősponttal elválasztva.
- A PRINT kulcsszó parancs formájában is helyettesíthető kérdőjellel.
- A számítógépen a tizedesvesszőt a tizedespont helyettesíti!

A megjegyzéseknek megfelelően tehát az előző három parancsot egyszerre is kiadhatjuk:

D=16.6 : D%=16.6 : PRINT D,D% [NL]

A programokban általában kényelmes bizonyos változókat egész típusúaknak definiálni a program elejére helyezett DEFINT utasítással.

A harmadik fejezetben tapasztalhattuk, hogy az egész típusú változókkal a számítógép gyorsabban dolgozik, mint a valós típusúakkal. Bonyolultabb programoknál nagyon fontos lehet, hogy minél gyorsabban lefussanak. (Ugyanakkor ne feledjük azt sem, hogy nem mindig az a legjobb program, amelyik a leggyorsabb...) Ezért érdemes kicsit részletesebben foglalkozni az egész típusú változókkal, az egész számok ábrázolásával és a velük végzett műveletekkel. Ezzel egyúttal megkezdjük a számítógéppel mint számológéppel való ismerkedésünket is.

Az egész számok ábrázolása két részből tevődik össze: a szám számjegyeiből, illetve az előjelből. A negatív előjelet mindig megjeleníti a gép a képernyőn, a pozitív előjelet viszont nem. A pozitív előjel helyét azonban (egy karakternyi helyet) minden esetben kihagyja a szám előtt:

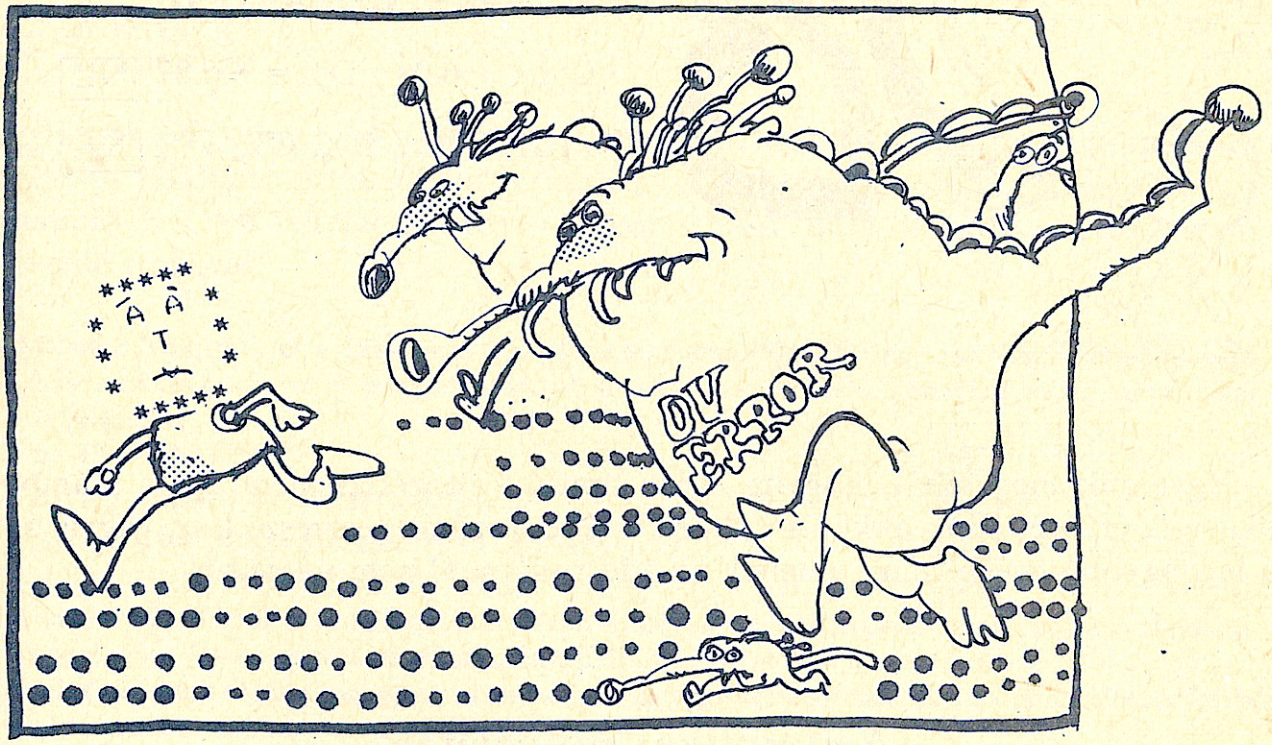
```
-31567
 23489
```

4.1 feladat: Készítsünk számláló programot, amelyik megkérdezi, hogy honnan kezdje a számlálást, hányasával számoljon, és növekedjenek vagy csökkenjenek az egész számok. Ügyeljünk arra, hogy az egymás után megjelenő számok ne túl gyorsan kövessék egymást! Használjunk egész típusú változókat!

Futtassuk az előbbi programot a következő adatokkal:

Kezdőérték	Lépésköz
Ø	+1
Ø	−1
−111	+22
743	−89
1Ø142	+51ØØ
−13446	−3342

Ne essünk kétségbe, ha OV Error hibajelzéssel áll le a futás!... Ez azt jelenti, hogy az egész típusú változók lehetséges értékeinek korlátait túlléptük. Ezt a jelenséget *túlcsordulás*nak nevezik.



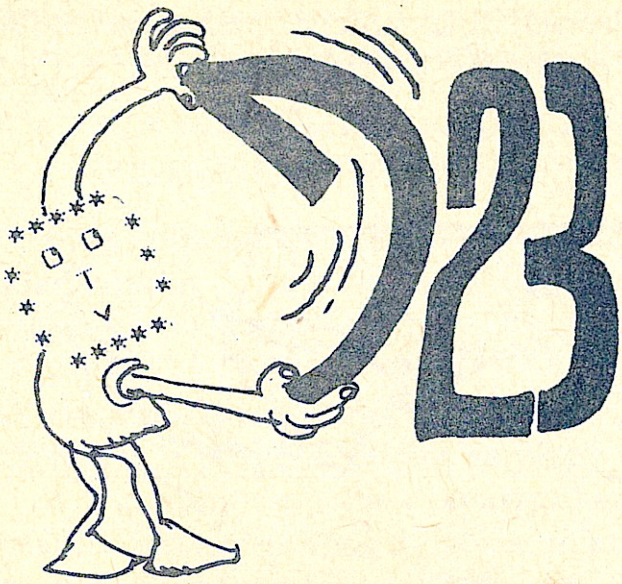
4.2 feladat: Az előző program segítségével pontosan határozzuk meg az egész típusú változók lehetséges értékeinek korlátait!

Egészrész

Vizsgáljuk meg, mi történik a következő parancs hatására:

SET(12Ø.98,1Ø.78) [NL]

Annak ellenére, hogy a SET utasítás argumentumában nem egész számok szerepelnek, ez a parancs mégis végrehajtódik. De vajon melyik pont gyulladt ki ennek hatására? Ezt saját magunk is eldönthetjük, ha kitalálunk hozzá egy ügyes kis programot!



Azt, hogy a számítógép hogyan „kerekít”, igen egyszerűen megállapíthatjuk a következő parancsok alapján:

A%=4.78 : B%=4.23 [NL]

C%= −6.Ø9 : D%= −6.87 [NL]

PRINT A%, B%, C%, D% [NL]

Jegyezzük meg tehát, hogyha egész típusú változónak nem egész számot akarunk értékül adni, akkor a változó értéke a szám egészrésze lesz, vagyis az a legnagyobb egész szám, amelyik az adott számnál nem nagyobb.

Aritmetikai műveletek

A változókkal és a számokkal *aritmetikai művelet*eket végezhetünk. Az összeadás és a kivonás jelölése a szokásos (+, −), ezekkel a műveletekkel már találkoztunk. A szorzás jele a csillag (*), az osztásé a törtvonal (/), a hatványozásé pedig a felfelé mutató nyíl (↑), de egyes géptípusoknál ezt a szögletes nyitózárójel ([ ) helyettesíti.

Az aritmetikai műveletek alkalmazására két példaprogramot mutatunk. Tanulmányozzuk át ezeket, fejtsük meg működésüket, majd ezután ellenőrzésül futtassuk le a programokat!

4.1 példaprogram

```
10 CLS
20 PRINT"ADJ MEG EGY EGESZ SZAMOT !
         EN ELDONTOM, HOGY HETTEL OSZTHATO-E." : INPUT A%
30 B%=A%/7 : B%=A%-7*B%
40 PRINT A% ;
50 IF B%<>0 THEN PRINT" NEM "; : REM B% ERTEKE NEM NULLA
60 PRINT" OSZTHATO HETTEL"
70 PRINT : PRINT
80 GOTO 20
```

4.2 példaprogram

```
1Ø DEFINT B,N
2Ø CLS
3Ø FOR N=1 TO 1Ø
4Ø    B=(2[N+3*N)/1Ø
5Ø    FOR X=Ø TO B : SET(X,3*N) : NEXT X
6Ø NEXT N
7Ø GOTO 7Ø
```

Felhívjuk a figyelmet a 4.1 példaprogram 3Ø-as, illetve a 4.2 példaprogram 4Ø-es sorában található értékadó utasításokra! A 4.1 példaprogram 5Ø-es sorában szereplő B< >Ø feltétel pontosan akkor IGAZ, ha a B változó értéke nem egyenlő nullával.

4.3 feladat: Végezzük el a következő műveleteket a számítógéppel parancs üzemmódban, majd hasonlókkal gyakoroljuk az egész számokkal és az egész típusú változókkal végzett aritmetikai műveleteket:

a) PRINT 23Ø364/9 [NL]

b) A%=9.2*81.42 : PRINT A% [NL]

c) A%=9.2 : B%=81.42 : PRINT A%*B% [NL]

d) A%=9.2 : A%=A%*81.42 : PRINT A% [NL]

4.4 feladat: Készítsünk olyan programot, amelyik megadja egy (a program által beolvasott) pozitív egész szám összes osztóinak számát, és kiírja az osztókat is!

4.5 feladat: Írjunk olyan programot, amelyik beolvas négy egész számot, és ezek közül először kiírja a párosakat, majd ezután a páratlanokat is!

A prioritás

A számítógép az aritmetikai műveletek elvégzésénél egy bizonyos műveleti sorrendet tart be az ún. *prioritási sorrend*nek megfelelően. Vizsgáljuk meg az alábbi értékadó utasítást!

$$Z = 5+9*13-3 [4/9$$

Itt először a hatványozást kell elvégezni, ezután a szorzást és az osztást, és végül az összeadást és a kivonást:

$$Z=5+9*13-81/9$$
$$Z=5+117-9$$
$$Z=113$$

Ez a sorrend azonban zárójelezéssel befolyásolható. Ekkor a számítógép először a zárójelen belüli műveleteket végzi el (prioritási sorrendben!).

4.6 feladat: Mennyi lesz a K% változó értéke, ha

```
a)  K%=456-67+18*131/12-42
b)  K%=456-(67+18)*131/12-42
c)  K%=(456-67+18)*131/12-42
d)  K%=456-67+18*131/(12-42)
e)  K%=(456-67+18)*131/(12-42)
f)  K%=456-(67+18)*131/(12-42)
g)  K%=456-(67+18)*(131/12-42)
```

Az aritmetikai műveletek közül a számítógép – ha azt zárójelek nem befolyásolják – először mindig a hatványozást, másodszor az előjelváltást, azután a szorzást és az osztást, végül az összeadást és a kivonást hajtja végre. Az azonos prioritású műveletek elvégzésénél a „balról-jobbra" szabály érvényes.

Találkozhatunk olyan kifejezésekkel, amelyekben aritmetikai és logikai műveletek, valamint relációk is szerepelnek. Az

X [3<6/X AND X<1Ø

kifejezés értéke például az X=1 értékre IGAZ, az X=3 értékre pedig HAMIS, hiszen az első esetben az 1<6 és az 1<1Ø egyenlőtlenség is teljesül, de a második esetben a 27<2 nem. Láthatjuk tehát, hogy ilyen esetben először az aritmetikai, majd azután a logikai műveletek kerülnek végrehajtásra. Vizsgáljuk még meg a *logikai műveletek* egymás közötti *prioritási sorrendjét* is! Készítsünk olyan programot, amelynek segítségével el tudjuk dönteni, hogy a következő utasításban szereplő logikai műveleteket milyen sorrendben végzi el a számítógép:

IF X<54 AND Y>5 OR X>63 AND Y>5 THEN 5Ø

Bizonyára rájöttünk arra, hogy a számítógép először az AND, és utána az OR logikai műveleteket hajtja végre, függetlenül azok előfordulási sorrendjétől. Azt mondjuk tehát, hogy az AND prioritása nagyobb, mint az OR-é. Ha zárójelek nem befolyásolják, akkor az azonos prioritású logikai műveletek elvégzésénél is a „balról-jobbra" szabály érvényes. Ha zárójeleket alkalmazunk, akkor először a zárójelen belüli kifejezéseket értékeli ki a számítógép. A fenti utasítás tehát egyenértékű az

IF (X<54 AND Y>5) OR (X>63 AND Y>5) THEN 5Ø

illetve az

IF (X<54 OR X>63) AND Y>5 THEN 5Ø

utasítással.

Végezetül, az *általános prioritási sorrend* a következő:

ELŐSZÖR: hatványozás
MÁSODSZOR: előjelváltás
HARMADSZOR: szorzás, osztás
NEGYEDSZER: összeadás, kivonás
ÖTÖDSZÖR: relációk (<, >, =)
HATODSZOR: logikai műveletek

4.7 feladat: Hol a hiba?

```
10 DEFINT X,Y,Z
20 Z=4
30 X=150/2 : Y=(7[2)*2
40 X=(X*Y*Z):3 : PRINT X
50 Z=Z+1 : GOTO 30
```

4.8 feladat: Hol a hiba?

```
10 CLS
20 INPUT"A NEGYZET OLDALA";X
30 PRINT"A NEGYZET KERULETE ES TERULETE =";4X;XX
40 PRINT/GOTO20
```

Gyakorlatok a negyedik fejezet anyagához

4.1 gyakorlat: Gyűjtsuk ki a képernyőn az összes olyan grafikus pontot, amelynek koordinátái négyzetszámok!

4.2 gyakorlat: Készítsünk programot, amelyik bekér öt lottószámot, majd kirajzol egy lottószelvényt, és azon bejelöli a megadott számokat!

4.3 gyakorlat: A képernyőn megjelenik egy négyzet. Ha elküldjük a 0 és a K számot, akkor a számítógép kirajzol egy olyan négyzetet, amelynek oldalai K-ad részei az előző négyzet oldalainak, ha pedig az 1 és a K számot, akkor az oldalak hosszát K-szorosára változtatja. Írjunk ilyen programot!

4.4 gyakorlat: A képernyőn megjelenik egy 4 × 4-es sakktábla, és annak egyik mezőjében egy L betű. A sakktábla mezőit 1-től 16-ig megszámoztuk. Készítsünk olyan programot, amelyik bekéri valamelyik mező sorszámát, és az L betűt elhelyezi erre a mezőre!

4.5 gyakorlat: Írjunk olyan programot, amelyik bekér egy számot, majd kiírja annak −1-szeresét, tízszeresét, négyzetét és ötödik hatványát! Használjunk egész típusú változókat, és igyekezzünk elkerülni a túlcsordulást!

4.6 gyakorlat: Készítsünk olyan programot, amelyik bekér három egész számot, és ezek rögzített sorrendje mellett elvégzi a négy alapműveletet az összes lehetséges módon, majd ki is írja az eredményeket!

4.7 gyakorlat: Készítsünk olyan programot, amelyik bekér egy pozitív egész számot (N), majd kiírja az első N páratlan szám, illetve az első N köbszám összegét!

4.8 gyakorlat: Írjunk olyan programot, amelyik bekér egy pozitív egész számot (N), majd ezután beolvas N darab számot, és kiírja ezek átlagát!

4.9 gyakorlat: Írjunk olyan programot, amelyiknek ha megadjuk egy kocka élhosszát, akkor kiírja egy táblázatba rendezve a kocka felszínét, térfogatát és éleinek összhosszúságát!

4.10 gyakorlat: Készítsünk olyan programot, amelyik kiszámolja egy megadott polinom helyettesítési értékeit a kívánt helyeken!

4.11 gyakorlat: Képzeljük magunkat egy, a Föld felé haladó űrhajó pilótájának! A számítógép képernyőjén leolvashatjuk a rendelkezésre álló üzemanyag mennyiségét, a Földtől való távolságot és az űrhajó sebességét. Ha a számítógépnek megadjuk, hogy mennyivel kívánjuk módosítani az űrhajó sebességét, és azt, hogy mennyi idő teljen el így, akkor kiírja az ennek megfelelő értékeket. Cél az űrhajó Földre való sima leszállítása. Készítsük el az ennek megfelelő programot!

4.12 gyakorlat: Készítsünk olyan programot, amelyik egy másodperc értéket átvált nap, óra, perc, másodperc formára!

4.13 gyakorlat: Készítsünk olyan programot, amelyik egy kettes számrendszerbeli számot tízesbe, egy tízes számrendszerbeli számot pedig kettesbe vált át!

4.14 gyakorlat: Készítsünk olyan programot, amelyik két tetszőleges, beolvasott számra elvégzi a maradékos osztást!

4.15 gyakorlat: Írjunk programot, amelyik egy adott számig kiírja a prímszámokat!

4.16 gyakorlat: Készítsünk programot, amelyik öt beolvasott számot nagyság szerinti csökkenő sorrendben kiír!

4.17 gyakorlat: Írjunk olyan programot, amelyik két beolvasott szám legnagyobb közös osztóját és legkisebb közös többszörösét meghatározza!

4.18 gyakorlat: Készítsünk olyan programot, amelyik két adott szám közötti számok közül keres egy prímszámot!

4.19 gyakorlat: Készítsünk programot, amelyik két megadott szám közül megkeresi azt, amelyiknek több osztója van!

4.20 gyakorlat: Gyűjtsuk ki a képernyő összes olyan pontját, amelyek koordinátái 5 és 45 közé esnek, de 10 többszöröseivel, valamint a 11 és a 19 közötti számokkal nem egyenlőek!

A negyedik fejezet feladatainak egy lehetséges megoldása

4.1 feladat

```
10 DEFINT K,S,X
20 CLS
30 INPUT"HONNAN KEZDJEM A SZAMLALAST ";K
40 PRINT : INPUT"HANYASAVAL SZAMLALJAK ";S
50 PRINT : INPUT"    NOVEKEDJENEK A SZAMOK (+1) ,
   VAGY CSOKKENJENEK (-1) ";X
60 IF X=1 OR X=-1 THEN S=S*X ELSE 50
70 PRINT K :
   FOR X=1 TO 100 : NEXT
80 K=K+S :
   GOTO 70
```

4.2 feladat

Az egész típusú változók lehetséges legkisebb értéke: −32768,
lehetséges legnagyobb értéke pedig: 32767.

4.3 feladat

a) 25596 *b)* 749 *c)* 729 *d)* 732

4.4 feladat

```
10 DEFINT I,J,M,N
20 CLS
30 INPUT"KEREK EGY POZITIV EGESZ SZAMOT ! , N= ";N
40 IF N<1 THEN GOTO 30
50 PRINT : PRINT N;" OSZTOI A KOVETKEZO SZAMOK :" : PRINT
60 J=1 : I=0
70 M=N/J : IF N=M*J THEN 80 ELSE 90
80 IF J*J=N THEN I=I+1 : PRINT J : GOTO 100
          ELSE I=I+2 : PRINT J;M
90 J=J+1 : IF J*J<N+1 THEN 70
100 PRINT : PRINT"TEHAT"; N;" OSZTOINAK SZAMA ";I
```

4.5 feladat

```
10 DEFINT I,J
20 CLS
30 PRINT : PRINT : PRINT"KEREK NEGY EGESZ SZAMOT !" :
   INPUT I1,I2,I3,I4 : PRINT
40 J=I1/2 : J1=I1-2*J : J=I2/2 : J2=I2-2*J :
   J=I3/2 : J3=I3-2*J : J=I4/2 : J4=I4-2*J
50 IF J1*J2*J3*J4=1 THEN PRINT"MINDEN SZAM PARATLAN :";
   I1;I2;I3;I4 : PRINT : GOTO 30
60 IF J1+J2+J3+J4=0 THEN PRINT"MINDEN SZAM PAROS :";
   I1;I2;I3;I4 : PRINT : GOTO 30
70 PRINT"PAROS SZAMOK :";
80 IF J1=0 THEN PRINTI1;
90 IF J2=0 THEN PRINTI2;
100 IF J3=0 THEN PRINTI3;
110 IF J4=0 THEN PRINTI4;
120 PRINT : PRINT"PARATLAN SZAMOK :";
130 IF J1=1 THEN PRINTI1;
140 IF J2=1 THEN PRINTI2;
150 IF J3=1 THEN PRINTI3;
160 IF J4=1 THEN PRINTI4;
170 GOTO 30
```

4.6 feladat

a) 543 *b)* −514 *c)* 4401 *d)* 310 *e)* −1778 *f)* 827 *g)* 3098

4.7 feladat

Az osztás jele nem a kettőspont, hanem a törtvonal (40-es sor).

A Z = 13 esetben a számolás eredménye még csak 31850, de Z = 14 esetén már 34300 lenne, ami túlcsordulást eredményez, hiszen egész típusú változókkal dolgozunk.

4.8 feladat

A szorzás jelét, a csillagot mindig ki kell tenni!

A 40-es sorban a kettőspont helyett törtvonal van.

Ötödik fejezet

TOVÁBB SZÁMOLUNK

A valós típusú változók

Ha egy programban valamely változónév után nem használunk semmiféle típusjelet, és a programban sehol sincsen erre vonatkozó típusdefiníció, akkor ezt a változót a gép automatikusan valós típusúként kezeli. A teljesség kedvéért megemlítjük, bár gyakorlati jelentősége kicsi, hogy a *valós típusú változó*kat is definiálhatjuk a változónév után írt felkiáltójellel vagy pedig a **DEFSNG** utasítással.

5.1 feladat: Alakítsuk át a 4.1 feladatra írt programot úgy, hogy alkalmas legyen nem egész valós számokkal történő műveletek elvégzésére is!

Ha az 5.1 feladat programját Ø vagy 1 ØØØ ØØØ (vagy −1 ØØØ ØØØ) körüli értékekkel futtatjuk, akkor érdekes jelenséget figyelhetünk meg. A valós típusú változók értékeit a gép kétféle formában írja ki. Az egyik a megszokott forma, például:

$$55.6,\ -2.467,\ 52769.3,\ \ldots$$

Ezt *fixpontos számábrázolás*nak nevezzük. (Felhívjuk a figyelmet arra, hogyha a tizedesponttól balra csak a Ø számjegy állna, akkor ezt a számítógép nem írja ki!).

A másik pedig a *lebegőpontos számábrázolás*. Ennek alapja az, hogy tetszőleges (nem nulla) valós szám $\pm\ v \cdot 1Ø^{\pm n}$ alakra hozható, ahol a v az ún. *mantissza*, 1-nél nem kisebb és 1Ø-nél kisebb fixpontos alakban írt valós szám, az n nemnegatív egész szám pedig a *karakterisztika*. Ezt az alakot a gép $\pm vE\pm n$ formában írja ki a képernyőre, ahol az E betű elválasztja a mantisszát a karakterisztikától.

Például a 0,0005; 10^6; $1{,}124 \cdot 10^{-13}$; $-9{,}99 \cdot 10^{22}$ ilyen alakban jelenik meg a képernyőn:

$$5E-Ø4;\ 1E+Ø6;\ 1.124E-13;\ -9.99E+22$$

Természetesen az, hogy a kiíratás melyik formában történik, nem attól függ, hogy a gépnek milyen a hangulata, hanem bizonyos határok között mindig

fixpontos, egyébként pedig lebegőpontos alakban jelennek meg a képernyőn az adatok. A számítógép tetszőleges valós típusú változó értékét lebegőpontos alakban ábrázolja a memóriájában, ugyanis így sokkal nagyobb számokkal is képes műveleteket végezni, mint fixpontos alakban tárolt számokkal. Ezek után éppen az a meglepő, hogy bizonyos számokat mégis fixpontos alakban ír ki. Ez a számítógépnek plusz feladatot jelent, a felhasználónak viszont nagy könnyebbséget, hiszen mi sokkal inkább a fixpontos ábrázoláshoz vagyunk hozzászokva.

5.2 feladat: Határozzuk meg, hogy a valós típusú változók esetében mely számoknál lép fel túlcsordulás! Ehhez jól felhasználható az 5.1 feladat programja!

Megjegyzés: Oldjuk meg a 4.3, illetve a 4.6 feladatokat valós típusú változók felhasználásával is!

A kiíratás pontossága

Az előző feladatok megoldása során tapasztalhattuk, hogy a számítógép a valós típusú változók értékeit legfeljebb hat értékes jegy pontossággal írja ki, ahol a hatodik már lehet kerekített érték is.

Találkozhatunk azonban olyan feladatokkal is, amelyek megoldásához nagyobb pontosságra van szükség. Erre ad lehetőséget a *dupla pontosságú valós típusú változók* használata. Ezek definiálása vagy a változónév után írt számjellel (#) vagy a **DEFDBL** utasítás segítségével történhet. A dupla pontosságú valós típusú változók értékeit a számítógép legfeljebb 16 értékes jegy pontossággal írja ki, ahol a tizenhatodik már lehet kerekített érték is.

Megjegyzés:
- Ügyeljünk arra, hogy a számítógép csak a négy alapműveletet végzi el dupla pontosságú értékekkel! (A hatványozást és a következőkben ismertetendő belső aritmetikai függvények helyettesítési értékeinek kiszámítását nem.)
- A HT iskolaszámítógép belső számábrázolása kettes számrendszerbeli. Az általunk megadott adatokat átváltja kettes számrendszerbe, a műveleteket itt végzi el, az eredményeket pedig ismét a tízes számrendszerben kapjuk meg. Ezek az átalakítások sok gépi pontatlanságot eredményezhetnek. Erre láthatunk néhány példát a következő parancsok segítségével:

```
A = 5/3:        ?A    [NL]
A# = 5/3:       ?A#   [NL]
A# = 5#/3:      ?A#   [NL]
A# = 5#/3#:     ?A#   [NL]

?3↑3            [NL]
?3↑3-27         [NL]
?3#↑3-27        [NL]
?3#↑3-27#       [NL]
```

A dupla pontosságú fixpontos kiíratás esetén csak annyi a különbség, hogy a kiírt szám több értékes jegyet tartalmazhat, a lebegőpontos kiíratásnál ezen kívül az E betű helyett a D betű jelenik meg. Például:

99999999999.12, –.ØØØØØØØ123, –1.7Ø1411834244556D+38,
3.56789Ø123D–31

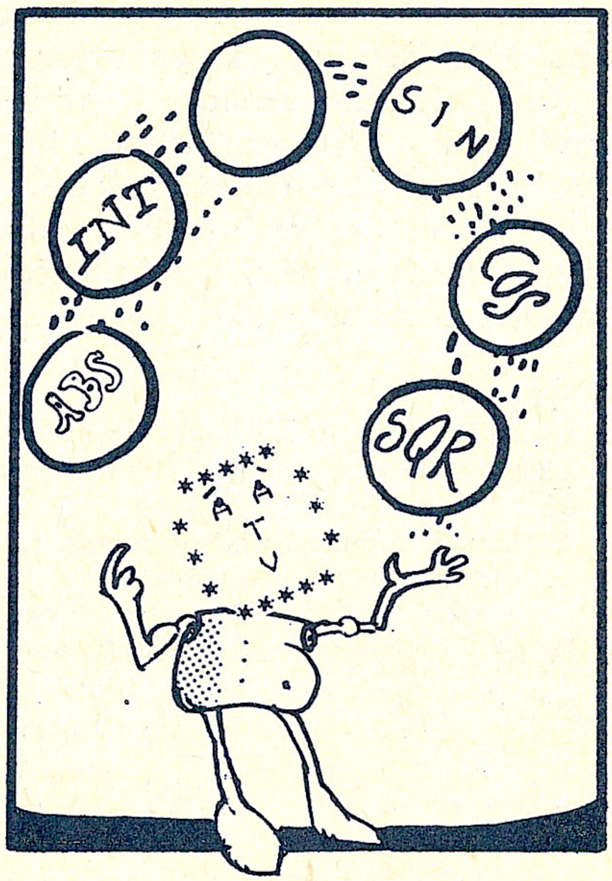
5.3 feladat: Készítsünk olyan programot, amelynek segítségével meg lehet határozni a következő egyenletnek a [2,3] intervallumba eső közelítő megoldását:

$$x^4-3x^2-1\text{Ø}=\text{Ø}$$

Belső aritmetikai függvények

Vannak olyan feladatok, amelyek megoldásánál nagy segítséget jelentenek az ún. *belső aritmetikai függvények*. Ezek közül néhány fontosabb a következő:

- abszolútérték függvény ABS
- négyzetgyök függvény SQR
- egészrész függvény INT
- szinuszfüggvény SIN
- koszinuszfüggvény COS
- exponenciális függvény EXP
- természetes alapú logaritmus függvény LOG



Futtassuk le a következő példaprogramot:

5.1 példaprogram

```
10 REM A BELSO ARITMETIKAI FUGGVENYEK HASZNALATA
20 CLS
30 REM ABS - ABSZOLUTERTEK FUGGVENY
40 PRINT"ABS(6.7)=";ABS(6.7),"ABS(-3.12)=";ABS(-3.12)
50 INPUT A : PRINT"ABS(";A;")=";ABS(A)
60 IF A=99 THEN CLS : GOTO 70 ELSE 50
70 REM SQR - NEGYZETGYOK FUGGVENY
80 PRINT"SQR(4)=";SQR(4),"SQR(178.35)=";SQR(178.35)
90 PRINT"SQR(X)       X>=0   !!!"
100 INPUT"X= "; A : PRINT"SQR(";A;")=";SQR(A)
110 IF A=99 THEN CLS : GOTO 120 ELSE 90
120 REM INT - EGESZRESZ FUGGVENY
130 PRINT"INT(4.2)=";INT(4.2),"INT(4,6)=";INT(4.6),
    "INT(-3.6)=";INT(-3.6),"INT(-3.1)=";INT(-3.1)
140 INPUT A : PRINT INT(A)
150 IF A=99 THEN CLS : GOTO 160 ELSE 140
160 REM SIN - SZINUSZ FUGGVENY
170 PRINT"SIN(3.14)=";SIN(3.14),"SIN(1.57)=";SIN(1.57)
180 PRINT : PRINT"A SZOGEK ERTEKET RADIANBAN KELL MEGADNI !" : PRINT
190 INPUT A : PRINT SIN(A)
200 IF A=99 THEN CLS : GOTO 210 ELSE 190
210 REM COS - KOSZINUSZ FUGVENY
220 PRINT"COS(3.14)=";COS(3.14),"COS(1.57)=";COS(1.57)
230 PRINT : PRINT"A SZOGEK ERTEKET RADIANBAN KELL MEGADNI !" : PRINT
240 INPUT A : PRINT COS(A)
250 IF A=99 THEN CLS : GOTO 260 ELSE 240
260 REM EXP - EXPONENCIALIS FUGGVENY
270 PRINT"EXP(0)=";EXP(0),"EXP(1)=";EXP(1)
280 INPUT A : PRINT EXP(A)
290 IF A=9 THEN CLS : GOTO 300 ELSE 280
300 REM LOG - TERMESZETES ALAPU LOGARITMUS FUGGVENY
310 PRINT"LOG(1)=";LOG(1),"LOG(2.71)=";LOG(2.71)
320 INPUT A : PRINT LOG(A)
330 IF A=99 THEN 20 ELSE 320
```

5.4 feladat: Készítsünk programot, amelyik kirajzolja az

$$f\colon [\emptyset, 125] \to \mathbf{R},\ x \mapsto \sqrt{16x}$$

függvény grafikonját!

5.5 feladat: Rajzoltassuk ki a szinuszfüggvénynek vagy valamely transzformáltjának a grafikonját a $[-2\pi, 2\pi]$ intervallumon! Törekedjünk arra, hogy a grafikon minél szebb legyen!

Megjegyzés: A negyedik fejezetben már felhívtuk a figyelmet arra, hogyha a SET utasítás argumentumában (vagyis a SET kulcsszó utáni zárójelek közötti részben) nem egész értékek állnak, akkor azoknak a számítógép az egészrészét veszi. Ennek megfelelően a számítógéppel való függvényábrázoláson azt értjük, hogy a függvény grafikonjának csak néhány pontját, és azokat is csak a megfelelő közelítéssel jelenítjük meg a képernyőn.

5.6 feladat: Végezzen egy pont harmonikus rezgőmozgást a képernyő két pontja között!

5.7 feladat: Hol a hiba?

```
10 CLS
20 FOR K=80 TO 0 STEP -1
30   Y=LOG(SIN(K*3.1416/32)
40   PRINT Y,
50 NEXT
```

5.8 feladat: Hol a hiba?

```
10 CLS
20 PRINT"ADJAL MEG EGY SZAMOT !   X=. "; : INPUT X
30 PRINT" AZ X SZAM LOGARITMUSA : ";
40 REM NAGYOBB-E NULLANAL A SZAM ? : IF X>0 THEN 50 ELSE
    PRINT"EZ NEM JO !" : GOTO 220
50 PRINT"LOG(";X;") =";LOG(X)
60 PRINT : GOTO 20
```

Gyakorlatok az ötödik fejezet anyagához

5.1 gyakorlat: Rajzoltassunk egy nagy kört a képernyőre!

5.2 gyakorlat: Jelenjen meg a képernyőn egy óra számlapja, és az óra nagymutatója, amelyik a negyedórákat mutatja!

5.3 gyakorlat: Jelenjen meg egy olyan pont a képernyőn, amelyik csillapított rezgőmozgást végez!

5.4 gyakorlat: Ábrázoljuk az 5.3 gyakorlatnak megfelelően mozgó pont kitérés-idő grafikonját!

5.5 gyakorlat: Készítsünk olyan programot, amelyik beolvas egy legfeljebb négyjegyű egész számot, és kiírja azt lebegőpontos alakban!

5.6 gyakorlat: Készítsünk olyan programokat, amelyek kiszámítják az alábbi egyenletek gyökét 3 tizedesjegy pontossággal:

a) $\sin x = \frac{x}{2}$;

b) $\cos^2 x = x^2$;

c) $2^x + 3^x + 4^x = 10$;

d) $e^x = 100 \sin x$.

5.7 gyakorlat: Írjunk olyan programot, amelyik egy háromszög egy oldala és a rajta fekvő két szög ismeretében kiszámolja a háromszög további adatait!

5.8 gyakorlat: Készítsünk olyan programot, amelyik valamely belső aritmetikai függvény helyettesítési értékeiből táblázatot készít!

5.9 gyakorlat: Készítsünk olyan programot, amelyik egy beolvasott N értéktől kezdve kiírja az

$$N \mapsto \left(1+\frac{1}{N}\right)^{N} \quad \text{és az} \quad N \mapsto \left(1+\frac{1}{N}\right)^{N+1}$$

sorozatok következő 10 tagját!

Az ötödik fejezet feladatainak egy lehetséges megoldása

5.1 feladat

A program megegyezik a 4.1 feladatra adott programmal, az ottani 1Ø-es sor nélkül.

5.2 feladat

A valós típusú változókba

$$-1.7Ø141E+38 \quad \text{és} \quad 1.7Ø141E+38$$

közötti számokat írhatunk.
(Az itt megadott értékekre már OV Error hibaüzenettel leáll a program futása.)

5.3 feladat

```
1Ø CLS
2Ø INPUT"X= ";X
3Ø Y=X[4-3*X*X-1Ø
4Ø PRINT"A HELYETTESITESI ERTEK :";Y
5Ø PRINT :
   GOTO 2Ø
```

5.4 feladat

```
1Ø REM TENGELYRAJZOLAS
2Ø CLS
3Ø FOR X=Ø TO 127 : SET(X,46) : NEXT X
4Ø FOR Y=Ø TO 47 : SET(2,Y) : SET(1,Y) : NEXT Y
5Ø REM A FUGGVENY ABRAZOLASA
6Ø FOR X=Ø TO 125
7Ø   Y=SQR(16*X) : Y=ABS(Y-47) : SET(X+2,Y)
8Ø NEXT X
9Ø GOTO 9Ø
```

5.5 feladat

```
1Ø REM TENGELYRAJZOLAS
2Ø CLS
3Ø FOR X=Ø TO 127 : SET(X,23) : NEXT X
4Ø FOR Y=Ø TO 46 : SET(63,Y) : NEXT Y
5Ø REM A GRAFIKON
6Ø FOR X=Ø TO 63
7Ø   Y=9*SIN(X*3.1416/32) : SET(63+X,23-Y) : SET(63-X,23+Y)
8Ø NEXT
9Ø GOTO 9Ø
```

5.6 feladat

```
1Ø CLS
2Ø DEFINT I
3Ø FOR I=Ø TO 63
4Ø   SET(63+SIN(I*3.14/32)*63,23) : RESET(63+SIN(I*3.14/32)*63,23)
5Ø NEXT I
6Ø GOTO 3Ø
```

5.7 feladat

A 3Ø-as sorban hiányzik a logaritmus függvény argumentumának záró zárójele.
A program futása ezután is hibajelzéssel fog leállni amikor a K változó értéke eléri a 63-at, mert ekkor a logaritmus argumentumában negatív érték lesz.

5.8 feladat

A 4Ø-es sorban levő feltételes utasítás sohasem hajtódik végre az előtte álló REM utasítás miatt.

Hatodik fejezet

JÁTSSZUNK A KARAKTEREKKEL!

RND

Számos feladat megoldásához, játék vagy számítógépes szimuláció elkészítéséhez nagy segítséget nyújthat az, ha ún. véletlenszámok állnak rendelkezésünkre. Ezeket egy *véletlenszámgenerátor* állítja elő, amit az **RND** függvény segítségével hívhatunk meg. Az RND(Ø) függvényutasítás hatására egy egynél kisebb nemnegatív véletlenszámot kapunk eredményül.

6.1 példaprogram

```
10 REM  V E L E T L E N S Z A M O K
20 CLS
30 X%=10*RND(0) : Y=RND(0) : Z#=RND(0)
40 PRINTX%,Y,Z#
50 REM FIGYELJ A KULONBOZO TIPUSU VALTOZOKRA !
60 FOR I=0 TO 2000 : NEXT
70 GOTO 30
```

6.1 feladat: Írjunk olyan játékprogramot, amelyben a gép „gondol" egy Ø és 1 közé eső számot, és a játékosnak ki kell találnia. Minden találgatás után a gép írja ki, hogy a játékos által megadott szám kisebb vagy nagyobb, mint a generált szám. Találatnak azt fogadjuk el, ha az eltérés Ø,ØØ1-nél kisebb.

Ha például 1ØØ-nál kisebb és 9Ø-nél nem kisebb véletlenszámot kívánunk előállítani, akkor használhatjuk a

$$9Ø + 1Ø*RND(Ø)$$

utasítást.

6.2 feladat: Készítsünk a 6.1 feladatéhoz hasonló programot, amelynek segítségével a játékosnak a gép által gondolt legfeljebb háromjegyű pozitív egész számot kell kitalálnia!

6.3 feladat: Módosítsuk a 6.2 feladat programját úgy, hogy a gép által gondolt legkisebb szám 15Ø, a legnagyobb pedig 159 lehessen!

A 6.2 feladatot egyszerűbben oldhatjuk meg az RND(K) függvényutasítás segítségével, ami egy K-nál nem nagyobb pozitív véletlen egész számot generál, ha a K pozitív szám. A K helyén szerepelhet tetszőleges aritmetikai kifejezés, amelynek aktuális értéke a pozitív K szám.



6.4 feladat: Módosítsuk a 6.1 feladatot úgy, hogy a gondolt szám 100 és 1000 közé essen, és a számítógép számolja azt, hogy a játékos hányadik kísérletre találta ki!

6.5 feladat: Vizsgáljuk meg az RND(K) függvényutasítás hatását, ha a K nem egész szám!

A véletlenszámok segítségével rajzolási technikánkat is tovább tudjuk fejleszteni, amire egyszerű példa a következő rövid program:

6.2 példaprogram

```
10 REM A KEPERNYO PONTONKENTI VELETLENSZERU KIGYUJTASA
20 CLS
30 SET(RND(128)-1,RND(48)-1) :
   GOTO 30
```

Karaktereket tartalmazó változók

A számítógép által használt változóknak van egy negyedik típusa is, a *karakteres* vagy ún. *string változók*. Ezek értéke legfeljebb 255 karakter hosszúságú karakterlánc lehet. A karakteres változókat a többi változótípusoktól a változónév után tett dollárjel ($) különbözteti meg. Például:

A$, BI$, KA$, ...

Ha nem akarjuk mindig kiírni a dollárjelet, akkor lehetőség van arra, hogy adott betűvel kezdődő változókat karaktertípusúként definiáljunk a **DEFSTR** utasítással.

6.3 példaprogram

```
10 REM K A R A K T E R E K
20 DEFSTR S,P
30 S="ABRAKA" : P="DABRA"
40 X4$="KAKUKKOSORA"
50 CLS
60 PRINTS;"-";P : PRINTX4$
```

A karaktertípusú változók értékadásánál az értékadó utasítás jobb oldalán vagy karaktertípusú kifejezés, vagy – kötelezően idézőjelek közé tett – karakterlánc állhat. Ez a karakterlánc nem tartalmazhat idézőjelet!

6.6 feladat: Rajzoltassunk négyzetet, amelynek oldalmérőszáma véletlenszám! Ha ezután elküldjük az "UJABB" szót, akkor a gép újabb négyzetet rajzoljon hasonlóan.

INKEY$

A számítógéppel való párbeszéd továbbfejlesztésére ad lehetőséget az INKEY$ utasítás.

6.4 példaprogram

```
10 REM PELDA AZ INKEY$ UTASITAS HASZNALATARA
20 CLS
30 SET(RND(128)-1,RND(48)-1)
40 IF INKEY$="T" THEN 30 ELSE 40
```

A 6.2 példaprogramot úgy módosítottuk, hogy a másodiktól kezdve a következő pont akkor gyullad ki, ha a [T] billentyűt lenyomjuk. Az INKEY$ utasítás segítségével a billenytűzetről legutoljára bejövő egy karakternyi információt vizsgálhatjuk meg a következőképpen:
Az A$=INKEY$ utasítás hatására az A$ változó értéke az utasítás előtt utoljára leütött billentyűnek megfelelő karakter lesz. (Ha a program inditása vagy a legutolsó INKEY$ utasítás végrehajtása óta nem ütöttünk le egyetlen billentyűt sem, akkor az üres karakter.) Az utasítást követően pedig az INKEY függvény értéke az üres karakter lesz.

Ez az utasítás a program futását nem szakítja meg, és hatására a képernyőn nem jelenik meg sem kérdőjel, sem a leütött billentyűnek megfelelő karakter.

Megjegyzés: A BREAK billentyű lenyomását nem tudjuk vizsgálni az INKEY$ utasítással, mert az a program futását megszakítja.

6.5 példaprogram

```
10 REM UJABB PELDA AZ INKEY$ UTASITAS HASZNALATARA
20 CLS
30 SET(RND(128)-1,RND(48)-1)
40 X$=INKEY$
50 IF X$="" THEN 40
60 IF X$=" " THEN 20 ELSE 30
```

Az első grafikus pont kigyújtása után a program aszerint folytatódik, hogy leütöttünk-e billentyűt, és ha igen, akkor az a SPACE volt-e. A 4Ø-es sort addig hajtja végre a számítógép, amíg egy billentyűt le nem ütünk. Ha ez a SPACE, akkor képernyőtörléssel és egy új pont kigyújtásával, ha bármelyik másik billentyű, akkor új pont kigyújtásával folytatódik a program.

Megjegyzés: Vegyük észre, hogy a számítógép megkülönbözteti a szóközt (SPACE) tartalmazó (" ") és az üres ("") karaktert! Az előbbi tartalmaz egy karaktert, míg az utóbbi egyet sem.

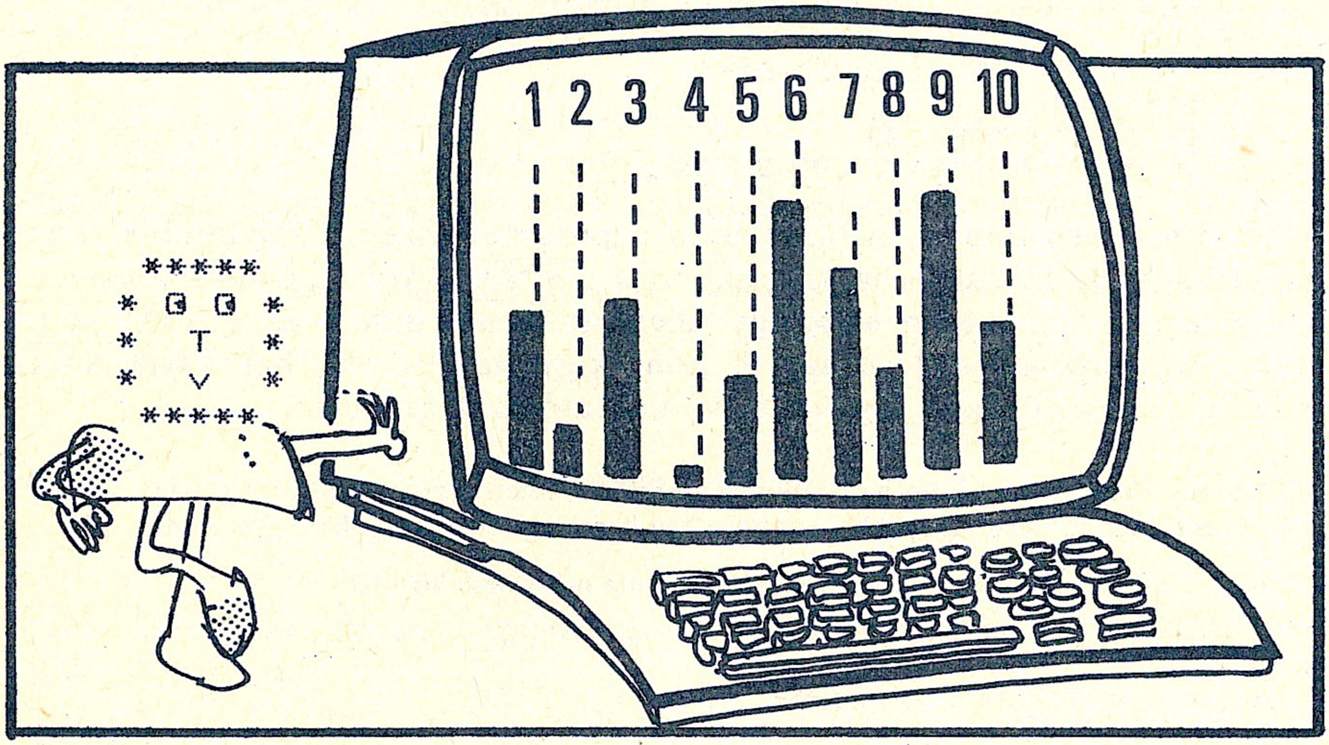
6.7 feladat: Mi a különbség a következő két program működése között?

```
10 REM ...... 1.PROGRAM
20 CLS
30 FOR X=0 TO 127 : SET(X,21) : NEXT X
40 A$=INKEY$
50 IF A$="" THEN 40
60 IF A$="T" THEN 20 ELSE 40
```

```
10 REM ...... 2.PROGRAM
20 CLS
30 FOR X=0 TO 127 : SET(X,21) : NEXT X
40 A$=INKEY$
50 A$=INKEY$
60 IF A$="" THEN 50
70 IF A$="T" THEN 20 ELSE 50
```

6.8 feladat: Készítsünk olyan programot, amelyben a számítógép gondol egy 1ØØ és 1ØØØ közé eső egész számot, amit nekünk ki kell találni. Miután kitaláltuk a gép által gondolt számot, a gép írja ki az ehhez szükséges próbálkozások számát, majd a billentyűzetről beadott információnak megfelelően vagy kezdődjék elölről (képernyőtörléssel) a játék, vagy fejeződjék be, és a gép "búcsúzzon el"!

6.9 feladat: Írjunk olyan programot, amelyik véletlenszerűen generál tíznél nem nagyobb természetes számokat! A program készítsen vonalas grafikont az egyes generált számokról olyan módon, hogy amikor egy újabb, például a 7-es szám generálódott, akkor a 7-eseknek megfelelő oszlop "tetején" egy újabb grafikus pont gyullad ki. A számokat addig állítsuk elő, amíg valamelyik oszlop a képernyő tetején levő számig nem ér. (Lásd az ábrát!)



6.10 feladat: Rajzoltassuk véletlenszerűen a számítógépet! Véletlenszerűen kiválasztott kezdőpontból, véletlenszerű (de legfeljebb 30 grafikus pontnyi) hosszúságú vonalat rajzoljon. Amikor egy szakaszt megrajzolt, a [T] billentyű leütésére folytassa a rajzolást az előbbi rendszer szerint, az [U] billentyű leütésére pedig képernyőtörléssel kezdje elölről.

6.11 feladat: Hol a hiba?

```
10 CLS
20 RND(0)=X
30 PRINTX,X*X,X*X*X : PRINT
40 FOR I=1 TO 300 : NEXT
50 GOTO 20
```

6.12 feladat: Hol a hiba?

```
10 CLS
20 SET(RND(127),RND(47)) : RESET(RND(127),RND(47))
30 IF INKEY="T" THEN 20 ELSE 30
```

6.13 feladat: Hol a hiba?

```
10 CLS :
   DEFINT S,X,Y
20 X=RND(130) : Y=X/2
30 IF X=Y*2 THEN 40 ELSE 20
40 SET(X,Y)
50 S=INKEY$
60 IF S="" THEN 60
70 IF S="T" THEN 20 ELSE 50
```

Gyakorlatok a hatodik fejezet anyagához

6.1 gyakorlat: Egy billentyű lenyomására villogjon három véletlenszerűen kigyújtott pont a képernyőn. Amikor ismét lenyomunk egy billentyűt, újabb három pont villogjon!

6.2 gyakorlat: Lenyomunk egy billentyűt, kigyullad egy pont. Ha ugyanazt a billentyűt ismételten lenyomjuk, akkor villogni kezd a pont, ha újabb billentyűt, akkor pedig egy új pont gyullad ki. Készítsük el ezt a programot!

6.3 gyakorlat: Mozogjon egy pont véletlenszerűen a képernyőn! Készítsünk ilyen programot!

6.4 gyakorlat: Készítsünk programot, amelyik egy hatszög csúcsait véletlenszerűen villogtatja!

6.5 gyakorlat: Írjunk olyan programot, amelyik tetszőleges billentyű lenyomására véletlenszerűen kiválasztott helyre rajzol egy háromszöget a képernyőn!

6.6 gyakorlat: Véletlenszerű időközönként szaladjanak fel vízszintes vonalak a képernyőn!

6.7 gyakorlat: Jelenjen meg két ablak a képernyőn, amiken a redőnyt véletlenszerűen le-föl húzogatják!

6.8 gyakorlat: Szimuláljuk számítógépen a következőt: 1ØØØ-szer dobunk egy dobókockával, és számoljuk, hogy az egyes számok hányszor fordultak elő.

6.9 gyakorlat: Készítsünk olyan programot, amelynek segítségével tetszőleges folytonos vonalat tudunk rajzolni a képernyőre! A rajzolást, vagyis az egymás utáni pontok kigyújtását négy billentyűvel vezéreljük: [F]-fel, [L]-le, [J]-jobbra, [B]-balra, és ha nem nyomunk le billentyűt, akkor arrafelé folytatódjék a rajzolás, amerre legutoljára irányítottuk.

6.10 gyakorlat: Jelenjenek meg véletlenszerűen a képernyőn az évszakok nevei!

6.11 gyakorlat: Készítsünk olyan programot, amelyik kiír két véletlenszámot a képernyőre! Ezután nekünk össze kell szorozni őket, majd az eredményről a számítógép döntse el, hogy helyes-e.

6.12 gyakorlat: Írjunk olyan programot, amelyik kijelöl a képernyőn egy osztást, amit nekünk el kell végezni. A számítógép ellenőrizze, hogy helyesen számoltunk-e! (A hányados mindig egész szám legyen!)

6.13 gyakorlat: Készítsünk olyan programot, amelyik azt méri, hogy milyen gyorsan tudunk egy adott jelre lenyomni egy billentyűt!

6.14 gyakorlat: Írjunk olyan programot, amelynek segítségével két játékos versenyezhet, hogy melyiknek jobbak a reflexei!

6.15 gyakorlat: Ketten játszanak egy dobókockával: *A* dob, de előzetesen *B* tippel, hogy *A* hányast fog dobni. Ha *B* tippje és az *A* által dobott szám megegyezik, akkor *B* ugyanannyi jutalompontot kap, mint amennyi a szám volt. Ha a tipp rossz volt, akkor a dobott és a tippelt szám közötti eltérésnek megfelelő büntetőpontot vonunk le pontszámaiból. Ezután cserélnek, *B* dob, *A* tippel stb. Az nyer, aki hamarabb ér el 1ØØ pontot.
Készítsük el a játéknak megfelelő programot!

A hatodik fejezet feladatainak egy lehetséges megoldása

6.1 feladat

```
10 CLS
20 X=RND(0)
30 PRINT"AZ X SZAMRA GONDOLTAM !" : PRINT :
   PRINT"TALALD KI X-ET !"
40 INPUT A
50 IF A-X>.001 THEN PRINT"X KISEBB" : PRINT : GOTO 40
60 IF X-A>.001 THEN PRINT"X NAGYOBB" : PRINT : GOTO 40
70 PRINT"     TALALT !" : PRINT :
   GOTO 30
```

6.2 feladat

```
10 CLS
20 DEFINT A,X
30 X=999*RND(0)+1
40 PRINT"AZ X SZAMRA GONDOLTAM !" : PRINT : PRINT"TALALD KI X-ET !"
50 INPUT A
60 IF X<A THEN PRINT"X KISEBB" : PRINT : GOTO 50
70 IF X>A THEN PRINT"X NAGYOBB" : PRINT : GOTO 50
80 PRINT"      TALALT !" : PRINT : PRINT : PRINT :
   GOTO 40
```

6.3 feladat

```
10 CLS
20 DEFINT A,X
30 X=150+10*RND(0)
40 PRINT"AZ X SZAMRA GONDOLTAM !          TALALD KI !!!"
50 INPUT A
60 IF X<A THEN PRINT" X KISEBB" : PRINT : GOTO 50
70 IF X>A THEN PRINT"X NAGYOBB" : PRINT : GOTO 50
80 PRINT"T A L A L T  !!!" : PRINT
90 STOP
```

6.4 feladat

```
10 CLS
20 DEFINT A,I,X
30 I=0 : X=100+RND(899)
40 PRINT"AZ X SZAMRA GONDOLTAM !" : PRINT :
   PRINT"TALALD KI X-ET !" : PRINT
50 I=I+1 :
   INPUT A
60 IF X<A THEN PRINT"X KISEBB" : PRINT :GOTO 50
70 IF X>A THEN PRINT"X NAGYOBB" : PRINT : GOTO 50
80 PRINT"     TALALT !" : PRINT : PRINT"EHHEZ ";I;
   " DARAB BROBALKOZASRA VOLT SZUKSEGED !" : PRINT :
90 GOTO 30
```

6.5 feladat

A számítógép a *K* egészrészét veszi. (Ezt a továbbiakban [*K*]-val jelöljük.) Ha [*K*] negatív, akkor FC Error hibajelzést kapunk. Ha nulla, akkor egy egynél kisebb nemnegatív véletlenszámot, ha pedig [*K*] pozitív, akkor *K*-nál nem nagyobb pozitív véletlen egész számot.

6.6 feladat

```
10 DEFINT A-Z
20 CLS
30 K=RND(44)
40 FOR X=0 TO 2*K
50   FOR Y=3 TO K+3 : SET(X,Y) : NEXT Y
60 NEXT X
70 INPUT A$
80 IF A$="UJABB" THEN 10 ELSE 70
```

6.7 feladat

Az első program futtatásakor, ha a vonal rajzolása közben lenyomjuk a [T] billentyűt, akkor a rajzolás befejezése után rögtön képernyőtörlés következik és kezdődik elölről a vonalrajzolás. A második program futásánál, ha a vonal rajzolása közben lenyomjuk a [T] billentyűt, akkor ez az információ a 6Ø-as sorban levő utasítás hatására elvész, így képernyőtörlés csak akkor történik, ha a vonal megrajzolása után nyomjuk le a szükséges billentyűt.

6.8 feladat

```
10 CLS
20 DEFINT A,I,X
30 I=0 : X=100+RND(899)
40 PRINT"AZ X SZAMRA GONDOLTAM !" : PRINT :
   PRINT"TALALD KI X-ET !" : PRINT
50 I=I+1 :
   INPUT A
60 IF X<A THEN PRINT"X KISEBB" : PRINT : GOTO 50
70 IF X>A THEN PRINT"X NAGYOBB" : PRINT : GOTO 50
80 PRINT"      TALALT !" : PRINT : PRINT"EHHEZ ";I;
   " DARAB PROBALKOZASRA VOLT SZUKSEGED ." : PRINT : PRINT
90 PRINT"FOLYTASSAM ?     (I/N)"
100 X$=INKEY$
110 IF X$="I" THEN 20
120 IF X$="N" THEN PRINT" VISZLAT !" ELSE 100
```

6.9 feladat

```
10 CLS
20 DEFINT X,Y
30 PRINT"        1    2    3    4    5    6    7    8    9    10"
40 X=10+10*RND(10) : Y=47
50 IF POINT(X,Y)=0 THEN SET(X,Y) : GOTO 40 ELSE Y=Y-1
60 IF Y>2 THEN 50
70 GOTO 70
```

6.10 feladat

```
10 DEFINT I,K,X,Y
20 CLS
30 X=RND(128)-1 : Y=RND(48)-1
40 SET(X,Y)
50 GOTO 120
60 X1=X : Y1=Y : I=RND(30) : K=RND(4)
70 IF K=4 THEN X=X+I : S=1 : IF X>127 THEN X=127
80 IF K=3 THEN X=X-I : S=-1 : IF X<0 THEN X=0
90 IF K=2 THEN Y=Y+I : S=1 : IF Y>47 THEN Y=47
100 IF K=1 THEN Y=Y-I : S=-1 : IF Y<0 THEN Y=0
110 IF K<3 THEN FOR I=Y1 TO Y STEP S : SET(X,I) : NEXT I
           ELSE FOR I=X1 TO X STEP S : SET(I,Y) : NEXT I
120 X$=INKEY$
130 IF X$="T" THEN 60
140 IF X$="U" THEN 20 ELSE 120
```

6.11 feladat

A 2Ø-as sorban az X változót az egyenlőtlenség bal oldalára kell tenni.

6.12 feladat

Az INKEY kulcsszó után kötelező a dollárjel kitevése. Érdemes megfigyelni, hogy a 2Ø-as sorban a RESET utasítás általában nem azt a pontot oltja ki, amelyiket ezen sor SET utasítása közvetlenül előtte kigyújtott! (Ez természetesen csak akkor hiba, ha a feladatunk egy olyan program készítése, amelyik a [T] billentyű lenyomására a képernyőn véletlen helyen felvillant egy pontot.)

6.13 feladat

Az 5Ø-, 6Ø-, 7Ø-es sorok szintaktikusan csak akkor lesznek helyesek, ha az S változót karaktertípusúként definiáljuk.

Ha a 2Ø-as sorban az X változó értékének 13Ø vagy 128 generálódik, akkor a program a 4Ø-es sorban hibával leáll.

Amennyiben a 6Ø-as sorra úgy érkezik a program végrehajtása, hogy előtte nem nyomtunk le egyetlen billentyűt sem, akkor itt a program végtelen ciklusba kerül. Ezért célszerű a THEN kulcsszó utáni címet 5Ø-re módosítani.

Hetedik fejezet

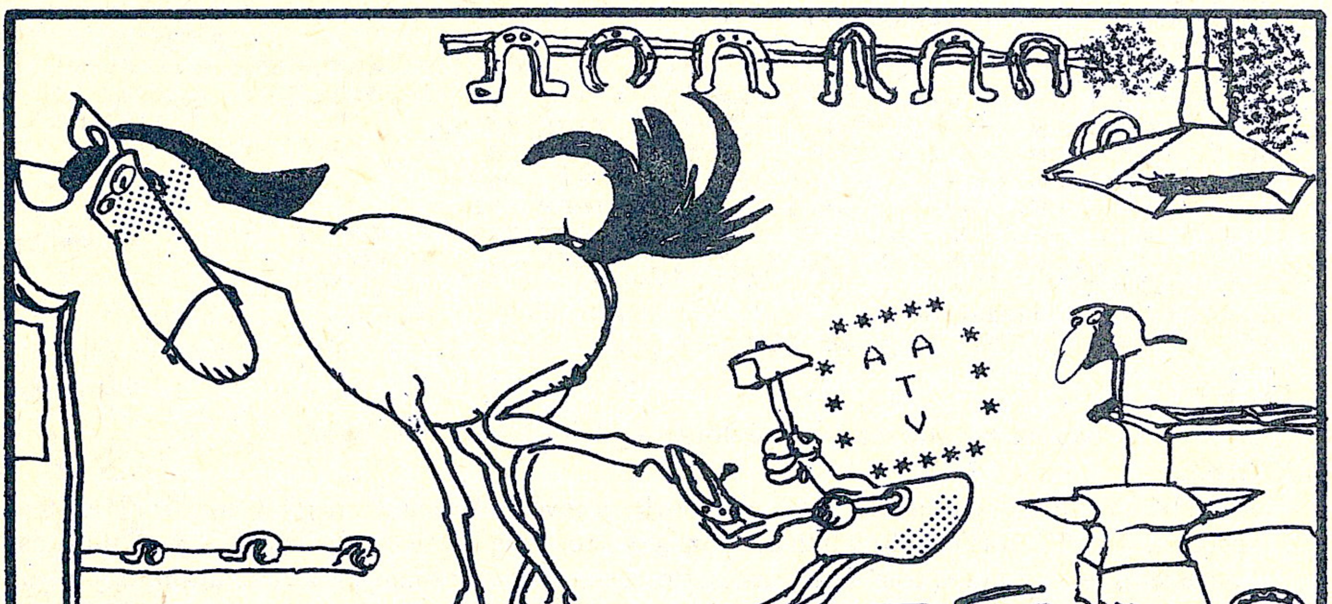
LÁNC, LÁNC, KARAKTERLÁNC...

A karakterláncok összeadása

Az előző fejezetben megismerkedtünk a karaktertípusú változókkal. Lehetőség van ezek tartalmának összeadására vagy más néven a *konkatenáció*ra, ami lényegében a karakterláncok összefűzésére szolgál. Így például az

A$="LO" : B$="PATKO" : C$=A$+B$ [NL]

parancs hatására a C$ változó tartalma "LOPATKO" lesz. Vigyázzunk! Ez az összeadás nem felcserélhető (nem kommutatív)! "LOPATKO" nem egyenlő "PATKOLO"!



7.1 feladat: Készítsünk olyan programot, amelyik beolvassa egymás után egy ötbetűs szó betűit fordított sorrendben (karakterenként), majd kiírja a szót!

CLEAR

Oldjuk meg a következő feladatot!

7.2 feladat: Írassunk ki a képernyőre 2ØØ darab csillag karaktert (*)! Próbálkozzunk több megoldással is!

Ezt a feladatot megoldhatjuk úgy is,
- hogy egy PRINT utasításba írunk 2ØØ darab csillag karaktert egymás mellé, vagy úgy,
- hogy 2ØØ darab PRINT"*"; utasítást hajtatunk végre a számítógéppel például egy FOR-ciklus segítségével, végül pedig úgy is,
- hogy egy karaktertípusú változót feltöltünk 2ØØ darab csillag karakterrel, például a következőképpen:

7.1 példaprogram

```
10 A$="*"
20 FOR I=1 TO 199
30   A$=A$+"*"
40 NEXT I
```

és ezen A$ változó tartalmát akarjuk kiíratni a képernyőre. Ekkor azonban OS Error hibajelzést kapjuk. Ugyanezt a jelenséget tapasztaljuk a következő példaprogram futtatásakor is, ha a C$ változónak olyan karakterláncot adunk értékül, amelyben ötvennél több karakter szerepel.

7.2 példaprogram

```
10 INPUT C$
20 PRINTC$
30 GOTO 10
```

A számítógép bekapcsolásakor automatikusan egy **CLEAR 5Ø** parancs hajtódik végre. Ez 5Ø karakternyi helyet biztosít a memóriában a karakterláncokkal való műveletek elvégzésére, illetve ezek eredményeinek tárolására. Ha a 7.2 példaprogram futtatása előtt kiadjuk a CLEAR 2ØØ [NL] parancsot, akkor a C$ változónak legfeljebb 2ØØ karakterből álló karakterláncot adhatunk értékül anélkül, hogy hibajelzést kapnánk. A 7.1 példaprogram esetében azonban a CLEAR 2ØØ [NL] parancs hatására biztosított hely kevés, mivel a karakterláncok összefűzéséhez még további helyekre van szükség. A CLEAR 4ØØ [NL] parancs már elegendő helyet biztosít.

Fontos megjegyezni azt is, hogy a CLEAR parancs (ami utasításként is használható) ezen kívül még törli az összes változó értékét és típusát is. Egy további hatásával a kilencedik fejezetben foglalkozunk.

STRING$, ASCII-kódok, CHR$, ASC

A 7.2 feladat megoldható a **STRING$** függvény segítségével is. Az

A$ = STRING$(2ØØ,"*")

utasítás hatására az A$ változó éppen 2ØØ darab csillag karakterrel töltődik fel.

Itt és a fejezet további részében az **s$** *vagy* idézőjelek közé tett karakterláncot (ami lehet egyetlen karakter vagy az üres karakter is, a karakterlánc pedig nem tartalmazhat idézőjelet), *vagy* karaktertípusú változót, *vagy* szövegkifejezést (pl. A$ + B$) jelöl. Az ***a, n, p*** betűk pedig *vagy* egy számot, *vagy* egy változót, *vagy* egy aritmetikai kifejezést (pl. A + 2*C[2) jelölnek. A vizsgált esetekben a számítógép mindig az *a*, *n*, *p* értékek egészrészét veszi, ezért a továbbiakban ezek egészrészéről szólunk anélkül, hogy ezt külön kiírnánk.

Ezeket felhasználva a STRING$ függvény meghívása kétféle módon lehetséges. A

STRING$(*p*,*s*$)

függvényutasítás az *s*$ első vagy egyetlen karakteréből álló *p* hosszúságú karakterláncot állít elő. (A *p* 256-nál kisebb nemnegatív egész szám, az *s*$ pedig itt nem lehet az üres karakter.)

7.3 példaprogram

```
10 A$="TATAMI" : B$="A "+A$ : C$="&"
20 D$=STRING$(11,"??") : K=7
30 PRINT D$; STRING$(5,C$); STRING$(3,A$)
40 PRINT STRING$(K,B$), STRING$(K-3,D$+B$)
```

A

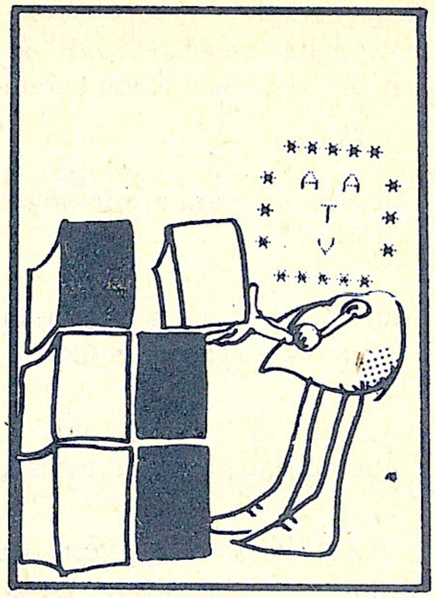
STRING$(*p*,*a*)

függvényutasításban a *p* értelmezése ugyanaz, mint előbb, az *a* számról pedig a következőket kell tudni:

A képernyőn megjeleníthető karakterekhez hozzárendeltek egy 256-nál kisebb nemnegatív egész számot, a karakter úgynevezett **ASCII-kódját.** A számítógép tetszőleges karaktert annak ASCII-kódja alapján azonosít. (ASCII-American Standard Code for Information Exchange: az információcsere egy szabványos amerikai kódja.)

Az első fejezetben már említettük, hogy a képernyőn 16 sorban, soronként 64 karakter jeleníthető meg egyszerre.

Egy karaktert tartalmazó mező még 6 részre, 6 grafikus pontra bontható.
Ezek mindegyike kigyújtható egy-egy SET utasítással. Ha egy karaktermező mind a 6 grafikus pontját egyszerre akarjuk kigyújtani, akkor a 191-es ASCII-kódot kell megadni.



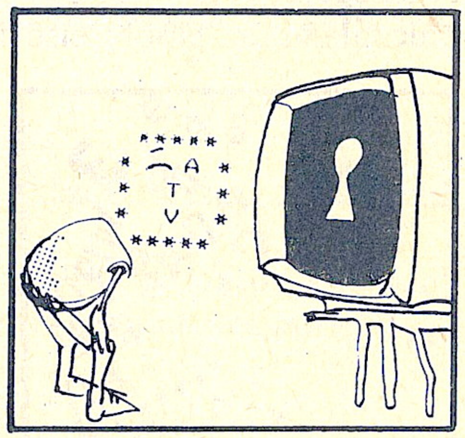
7.4 példaprogram

```
10 CLEAR 64
20 REM EGY SOR OSSZES KARAKTERMEZOJENEK KIGYUJTASA
30 D$=STRING$(64,191)
40 CLS
50 PRINT D$
```

7.3 feladat: Gyújtsuk ki a képernyő összes grafikus pontját egy másodpercnél rövidebb idő alatt!

Ez már lényeges előrelépést jelent a harmadik fejezetben megismert rajzolási sebességhez képest, de még ezt is lehet fokozni...

Érdekes lenne megtudni azt, hogy a különböző ASCII-kódoknak milyen karakterek felelnek meg.



A **CHR$** függvény ehhez nyújt nagy segítséget. A

CHR$(*a*)

függvényutasítás az *a* ASCII-kódhoz tartozó karaktert állítja elő.

7.4 feladat: Írassuk ki 32–191-ig az ASCII-kódokhoz tartozó karaktereket! Újabb karakter mindig egy billentyű lenyomására jelenjen meg.

Megjegyzések:

- Vizsgáljuk meg az ASCII-kódok jelentését 192–255-ig!
- A Ø–31-es ASCII-kódú karakterek az ún. *vezérlő karakterek*. Például a

PRINT CHR$(23)

utasítás hatására a számítógép egy sorba 64 helyett csak 32 karaktert ír ki szóközökkel. A

PRINT CHR$(3Ø)

utasítás hatására az aktuális kurzorpozíció sorában a kurzor utáni helyek törlődnek. (*Kurzor*nak nevezzük a képernyőn mozgó kis vízszintes vonalat.) A

PRINT CHR$(31)

utasítás hatására az aktuális kurzorpozíciótól kezdve a képernyő végéig minden törlődik.

Az ASC(*s*$) függvényutasítással megkapjuk az *s*$ első vagy egyetlen karakterének ASCII-kódját. Itt sem lehet az *s*$ az üres karakter!

7.5 feladat: Készítsünk olyan programot, amelyik bekér egy karaktert, majd kiírja az ASCII-kódját! Tetszőleges billentyű lenyomására megismétli ezt.

Karakterláncok összehasonlítása

Képzeljük el, hogy feladatul kapjuk nevek számítógéppel történő névsorba rendezését, azaz karakterláncokat kell összehasonlítani. Erre ezek ASCII-kódja segítségével van mód. Egy karakterlánc „nagyobb" valamely másik karakterláncnál, ha az előbbi első karakterének ASCII-kódja nagyobb a másik első karakterének ASCII-kódjánál. Ha a két karakterlánc elején azonos karakter(ek) van(nak), akkor balról az első különböző karakter kódja dönt. Ha egy karakterlánc egy másik karakterlánccal kezdődik, akkor a hosszabb a „nagyobb".



Az elmondottaknak megfelelően az alábbi állítások IGAZ-ak:

"KUTYA" < "MACSKA"
"PIRI" < "PIRIKE"
"WILMOS" < "ARANKA"
"B12" < "B52"

7.6 feladat: Készítsünk olyan programot, amelyik beolvas 3 szót, ábécésorrendbe rendezi, majd kiírja őket!

LEN, LEFT$, RIGHT$, MID$

Az *s*$ hosszát megkaphatjuk a **LEN** függvény segítségével.

7.5 példaprogram

```
10 A$="40" : B$="" : C$="RABLO"
20 PRINT LEN(A$), LEN(B$)
30 PRINT LEN(A$+B$+C$), LEN(A$+B$), A$+" "+C$
```

Az *s*$ első *p* darab karakterét a LEFT$(*s*$,*p*) függvényutasítással, az utolsó *p* darab karakterét pedig a RIGHT$(*s*$,*p*) függvényutasítással kaphatjuk meg. Értelemszerűen a *p* nem lehet negatív.

7.6 példaprogram

```
10 T$="LEVELIBEKA" : B$="" : C$="PETE" : L=5
20 PRINT LEFT$(T$,L+1), "-", RIGHT$(T$,4)
30 PRINT LEFT$(B$,3), RIGHT$(T$,12.5)
40 PRINT RIGHT$(T$+C$,LEN(C$)+4)
```

Lehetőség van egy karakterlánc valamely részének kiemelésére a **MID$** függvény segítségével. A

MID$(*s*$,*n*,*p*)

függvényutasítással az *s*$ *n*-edik karakterétől kezdődően *p* hosszúságú karakterláncot emelhetünk ki. A

MID$(*s*$,*n*)

függvényutasítással pedig az *s*$ *n*-edik karakterével kezdődő és az *s*$ további karaktereit tartalmazó karakterláncot kapjuk meg.

Fontos megjegyezni, hogy a karakterláncok karaktereinek számozása 1-től kezdődik és nem Ø-tól, ennek megfelelően az *n* pozitív. A *p* nemnegatív.

7.7 példaprogram

```
10 R$="RUHAFOGAS"
20 PRINT MID$(R$,5,3), MID$(R$,5)
30 PRINT MID$(R$,3,2)+" "+MID$(R$,5,3)
40 PRINT MID$(LEFT$("LOTTOHUZAS",5),2)
```

7.7 feladat: Készítsünk olyan programot, amelyik megkeresi két szó közös betűit!
7.8 feladat: Írjunk olyan programot, amelyik kiírja egy beolvasott szó szótagjainak számát!

VAL, STR$

A **VAL** függvénnyel karakterláncot numerikus értékké alakíthatunk át. Ha a vizsgált karakterlánc számmal kezdődik, akkor ezt a számot kapjuk eredményül, ha más típusú karakterrel, akkor nullát.

7.8 példaprogram

```
10 T$="KEREK PEREC"
20 PRINT VAL(T$), VAL("561"), VAL("101 KISKUTYA"), VAL("SKODA 125")
```

Numerikus értékeket karakterlánccá tudunk átalakítani a **STR$** függvény segítségével.

7.9 példaprogram

```
10 K=2.5 : L=-3.14
20 PRINT K; K; K,STR$(K); STR$(K); STR$(K) : PRINT
30 PRINT STR$(L), STR$(SQR(2*32)), STR$(2.5[2)
```

7.9 feladat: Hol a hiba?

```
10 CLEAR 500
20 I=1 : A$=$
30 FOR J=1 TO 8
40   I=I*2
50 NEXT
60 CLS
70 FOR J=1 TO I
80   A$=A$+$ : PRINT A$
90 NEXT
```

7.10 feladat: Hol a hiba?

```
10 DEFSTR N
20 INPUT N
30 PRINT STRING(ASC(N),CHR(ALMA+64) : GOTO 20
```

7.11 feladat: Hol a hiba?

```
10 CLS
20 INPUT "A SZO ";A$
30 FOR J=LEN(A$) TO 0 STEP -1
40 PRINT MID$(A$,J,LEN(A$)-J+1)
50 NEXT
60 STOP
```

Gyakorlatok a hetedik fejezet anyagához

7.1 gyakorlat: Készítsünk olyan programot, amelyik beolvas egy szót, majd kiírja a szó betűit fordított sorrendben!

7.2 gyakorlat: Írjunk programot, amelyik megállapítja, hogy egy beolvasott szövegben hány darab *A* betű van!

7.3 gyakorlat: Készítsünk olyan programot, amelyik beolvas egy szót, majd kiírja úgy, hogy minden *A* betűje helyett csillagok jelennek meg!

7.4 gyakorlat: Írjunk olyan programot, amelyik beolvas egy szöveget és kiírja azt szóközök nélkül!

7.5 gyakorlat: Megadunk egy szót csupa nagybetűvel. Készítsünk olyan programot, amelyik kiírja csupa kisbetűvel!

7.6 gyakorlat: Írassuk ki közvetlenül egymás mellé a négyzetszámokat 2ØØ-ig!

7.7 gyakorlat: Írjunk olyan programot, amelyik egy kettes számrendszerbeli számot tízesbe, egy tízes számrendszerbeli számot pedig kettesbe vált át!

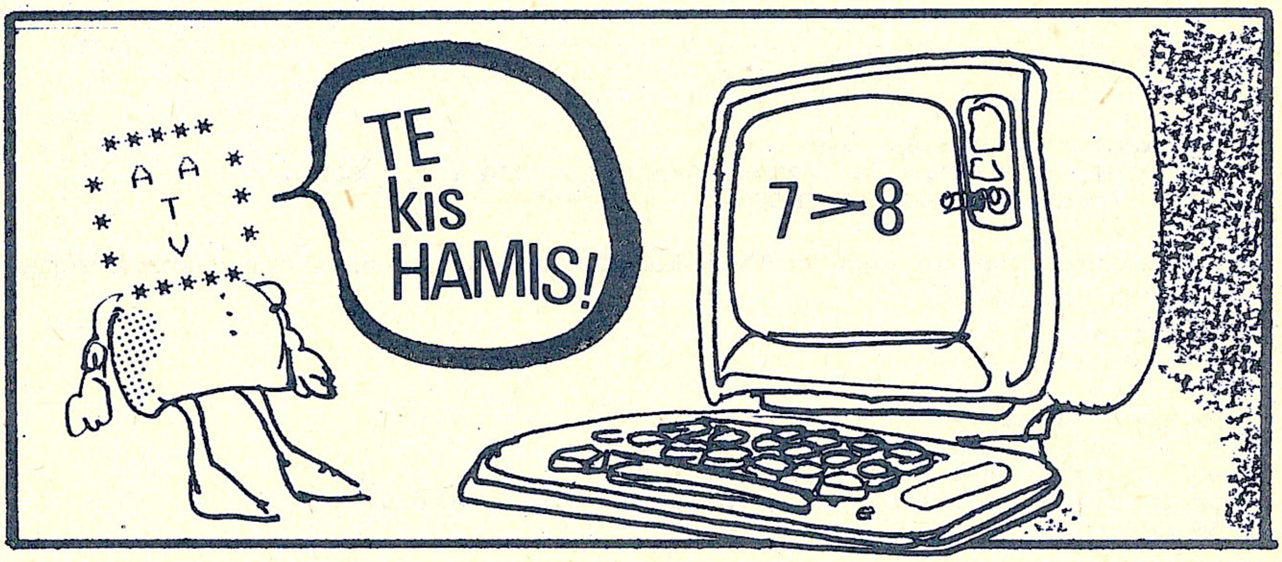
7.8 gyakorlat: Írassuk ki kétjegyű számok szorzatát! Ha a szorzat csak háromjegyű, akkor elé egy nullát írjon a számítógép.

7.9 gyakorlat: Írjunk olyan programot, amelyik beolvas tíz számot, majd ezeket folyamatosan, oszlopba rendezve kiírja úgy, hogy egy sorba pontosan öt szám*jegy* kerüljön! (Esetleg az utolsóba kevesebb.)

7.10 gyakorlat: A számítógépnek megadunk egy összeadást vagy szorzást a következő karakteres formában:

"3+4" , "5*6" .

Készítsünk olyan programot, amelyik kiírja e művelet eredményét!



A hetedik fejezet feladatainak egy lehetséges megoldása

7.1 feladat

```
10 DEFSTR A,B
20 CLS
30 PRINT"AZ OTBETUS SZO BETUI FORDITOTT SORRENDBEN ?" : PRINT
40 FOR I=1 TO 5 : INPUT B : A=B+A : NEXT I
50 PRINT A : PRINT
```

7.2 feladat

```
10 CLS
20 FOR I=1 TO 200 : PRINT "*"; : NEXT I
```

7.3 feladat

```
10 CLEAR 128
20 A$=STRING$(128,191)
30 FOR I=1 TO 8 : PRINT A$; : NEXT I
40 GOTO 40
```

(Előfordulhat, hogy ez a program egyes gépeken egy másodpercnél valamivel hosszabb idő alatt gyújtja ki a képernyő összes grafikus pontját.)

7.4 feladat

```
10 CLS
20 FOR I=32 TO 191
30   PRINT CHR$(I),
40   IF INKEY$="" THEN 40
50 NEXT I
```

7.5 feladat

```
10 CLS
20 INPUT "A KARAKTER ";A$
30 PRINT"AZ A$= ";A$;"  KARAKTER ASCII KODJA : "; ASC(A$)
40 IF INKEY$="" THEN 40 ELSE 10
```

Ha a szóközt tartalmazó karakter ASCII-kódját akarjuk megtudni, akkor a szóközt idézőjelek között kell megadni.

7.6 feladat

```
10 CLS
20 INPUT"A HAROM SZO "; A$, B$, C$
30 IF A$>B$ THEN D$=A$ : A$=B$ : B$=D$
40 IF C$>B$ THEN PRINT A$,B$,C$ : STOP
50 IF C$>A$ THEN PRINT A$,C$,B$ ELSE PRINT C$,A$,B$
60 END
```

7.7 feladat

```
10 CLS
20 INPUT "A KET SZO ";S$,Z$
30 J1=LEN(S$) : J2=LEN(Z$) : B$=" "
40 FOR K1=1 TO J1
50   FOR K2=1 TO J2
60     Q$=MID$(S$,K1,1)
70     IF Q$<>MID$(Z$,K2,1) THEN 120
80     L=0
90     L=L+1
100    IF Q$=MID$(B$,L,1) THEN 120
110    IF L=LEN(B$) THEN B$=B$+Q$ ELSE 90
120 NEXT K2,K1
130 PRINT : PRINT"A KOZOS BETUK : ";B$ : PRINT
```

7.8 feladat

```
10 CLS
20 Z$="AEIOU"
30 INPUT"A SZO ";S$
40 J1=LEN(S$) : J2=5 : B$=""
50 FOR K1=1 TO J1
60   FOR K2=1 TO J2
70     Q$=MID$(S$,K1,1)
80     IF Q$=MID$(Z$,K2,1) THEN B$=B$+Q$
90 NEXT K2,K1
100 PRINT : PRINT"A SZO "; LEN(B$); "  SZOTAGU" : PRINT
```

7.9 feladat

A karaktertípusú változók értékadásánál és a konkatenációnál a dollár karaktert idézőjelek közé kell tenni. Az *A*$ változó értéke legfeljebb 255 hosszúságú karakterlánc lehet.

7.10 feladat

A STRING és a CHR kulcsszavak után mindig ki kell írni a dollárjelet.

Ezután ez a program kiszámítja a beolvasott *N* karakterlánc első karakterének az ASCII-kódját. Amennyiben az ALMA nevű változó értéke Ø, akkor kiír az előző kód értékének megfelelő számú „@” karaktert.

7.11 feladat

A $J=\emptyset$ értékre a program hibajelzéssel leáll.

Nyolcadik fejezet

RUGALMASABB KÉPERNYŐCÍMZÉS

PRINT@

A könyv előző fejezeteiben többször említettük, hogy a számítógép képernyőjén 16 sorban soronként 64 karakter jeleníthető meg. A gépnek a

CLS : PRINT@5ØØ,"HAHO" [NL]

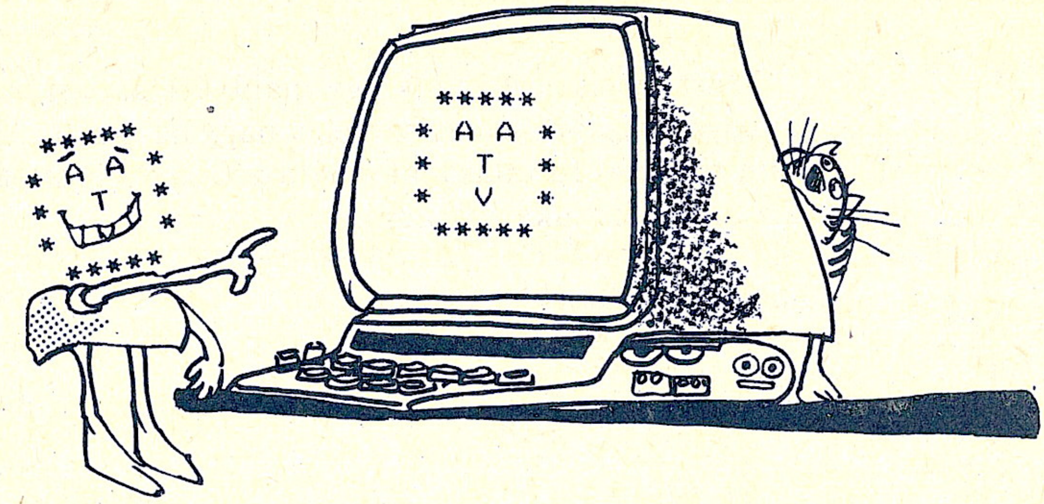
parancsot adva, a 7. sor 52. karakterpozíciójától kezdve kiíródik a "HAHO" felirat ($5ØØ = 7 \times 64 + 52$). A képernyő bal felső sarkában van a Ø. karakterpozíció, a nulladik sornak a következő karakterei az 1., 2., ... Ennek megfelelően az első sor nulladik karaktere a 64., a tizenötödik sor hatvanharmadik karaktere az 1Ø23. karakterpozíció. A **PRINT@I,*s*$** utasítással tehát a kurzort (ha az I nemnegatív egész szám) az I-edik karakterpozícióra állíthatjuk, majd innentől kezdve kiírhatjuk az *s*$-et. A kurzor ilyen módon való címzésével az I-edik karakterpozíció előtti helyek nem törlődnek.

Az utasításban szereplő I számra vagy a helyette álló aritmetikai kifejezés értékére $Ø \leq I < 1Ø24$ kell hogy teljesüljön, az *s*$-gel kapcsolatban pedig a hetedik fejezetben elmondottak érvényesek.

8.1 feladat: Vizsgáljuk meg a PRINT@I,s$ utasítás hatását, ha az I értéke nem egész szám!
8.2 feladat: Készítsünk olyan programot, amelyik valamilyen karakterekből álló keretben kiírja a nevünket és címünket!

A 8.2 feladat megoldása során bizonyára felvetődött az a gondolat, hogy a most megismert utasítás segítségével nemcsak feliratokat helyezhetünk el a képernyőn, hanem szabadabban készíthetünk karakterekből álló ábrákat is.

8.3 feladat: Rajzoltassuk ki könyvünk kísérő figuráját a képernyőre!



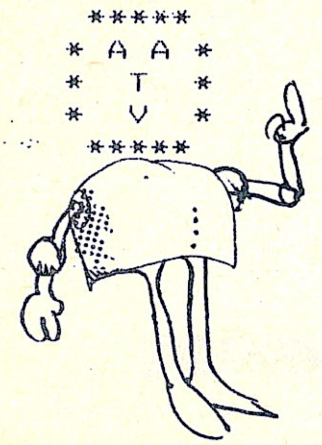
Tabulálás

A PRINT utasítás következő fajtája a táblázatok elkészítését könnyíti meg, amelynek alakja **PRINT TAB(K)**. A *K* aktuális értékének Ø és 255 között kell lennie. A számítógép *K* értékének 64-gyel való osztási maradékát veszi, és ha ez nagyobb, mint az aktuális PRINT utasítás sorában a kurzor pozíciója, akkor a kurzort az osztási maradéknak megfelelő karakterpozícióra állítja, ha kisebb, akkor a gép a TAB(*K*) utasítást figyelmen kívül hagyja.

8.4 feladat: Készítsünk programot, amelyik az alábbi táblázatot kirajzolja, és a hiányzó adatokat kitölti:

X	INT(X)	25*X	X*X	SQR(X)
Ø				
.5				
1				
1.5				
2				
2.5				
3				
3.5				
4				
4.5				
5				
5.5				
6				

DATA, READ, RESTORE



Most megismerkedünk egy újabb értékadási móddal, amelynek különösen akkor lehet nagy haszna, ha a program többszöri futtatásánál a változóknak mindig ugyanazt az értéket akarjuk adni.

8.1 példaprogram

```
1Ø REM A DATA ES A READ UTASITASOK HASZNALATA
2Ø CLS
3Ø READ X,Y,Z,A$,B$,C$
4Ø PRINT X,Y,Z : PRINT A$;B$;C$
5Ø DATA 14,3.14,-2,A,"   ",KUKAC
```

A 3Ø-as és az 5Ø-es sorok együttesen egyenértékűek az alábbival:

3Ø X = 14 : Y = 3.14 : Z = −2 : A$ = ”A” : B$ = ” ” : C$ = ”KUKAC”

Ügyeljünk arra, hogy a **DATA** utasításban felsorolt, vesszővel elválasztott adatok típusai megegyezzenek a **READ** utasításban a nekik sorrendben megfelelő változók típusaival! A DATA utasítást a programban bárhol elhelyezhetjük. Egy programban több DATA és READ utasítás is szerepelhet:

8.2 példaprogram

```
1Ø CLS
2Ø READ X,A$,B$
3Ø READ Y,Z,C$
4Ø PRINT X,Y,Z : PRINT A$;B$;C$
5Ø DATA 14,A
6Ø DATA "   ",3.14,-2,KUKAC
7Ø DATA "EZT MAR NEM HASZNALJA FEL A PROGRAM !"
```

Ha egy DATA utasításban kevesebb adat van, mint ahány változó a READ utasításban, akkor a gép a következő DATA utasítás adatait kezdi beolvasni. Amennyiben ilyet nem talál, akkor a program hibajelzéssel leáll.

A DATA utasításban csak akkor kell a karakterláncot idézőjelbe tenni, ha az szóközzel (” ”) kezdődik vagy végződik. Az idézőjelek közé tett karakterlánc nem tartalmazhat idézőjelet!

8.5 feladat: Oldjuk meg a 8.3 feladatot DATA és READ utasítások segítségével is!

8.6 feladat: Írassunk ki a képernyőre egy legalább két sorból álló szöveget hasonlóan, mint az a televízió BETŰREKLÁM-jában látható!

8.7 feladat: Rajzoltassunk ki a képernyőre egy egyszerű ábrát, majd lássuk el feliratokkal! Az adatokat DATA utasításokkal tároljuk.

8.8 feladat: Készítsünk programot, amelyik a képernyőn egy bábut jelenít meg és annak valamelyik testrészét mozgatja!

Egy programban szerepelhet a **RESTORE** utasítás is. Ennek hatására a következő READ utasításnál a legelső DATA utasítás első elemét kezdi beolvasni a program.

8.3 példaprogram

```
10 CLS
20 RESTORE
30 READ A$
40 IF A$="VEGE" THEN STOP
50 PRINT A$ : PRINT
60 X$=INKEY$
70 IF X$="T" THEN 30
80 IF X$="E" THEN 10 ELSE 60
90 REM A HET TORPE
100 DATA SZUNDI,SZENDE,HAPCI,MORGO,TUDOR,KUKA,VIDOR
110 DATA VEGE
```

8. 9. feladat: Hol a hiba?

```
10 CLS
20 FOR I=0 TO 15
30    PRINT@I*64,I : PRINT@I*64+3;". SOR"
40 NEXT
50 GOTO 50
```

8.10 feladat: Hol a hiba?

```
10 CLS
20 FOR X=1 TO 10
30    READ Y :
        PRINTX*64+X,Y
40 NEXT
50 DATA A,B,C,D,E,F,G,H,I
```

8.11 feladat: Hol a hiba?

```
10 CLS
20 FOR K=1 TO 16
30    PRINT@K*64-6,K*64-1;
40 NEXT
50 GOTO 50
```

Gyakorlatok a nyolcadik fejezet anyagához

8.1 gyakorlat: Jelenjen meg a képernyő különböző helyein a "VILLOG" felirat!

8.2 gyakorlat: Készítsünk olyan programot, amelyik megkérdezi, hogy hova és milyen szöveget írjon ki a képernyőre, és ezután végre is hajtja azt!

8.3 gyakorlat: Rajzoltassunk egyszerű figurát karakterekből!

8.4 gyakorlat: Oldjuk meg a 8.3 gyakorlatot SET és DATA utasítások segítségével!

8.5 gyakorlat: Rajzoltassuk ki néhány kémiai vegyület szerkezeti képletét a képernyőre!

8.6 gyakorlat: Rajzoltassunk olyan térképet a képernyőre, amelyen Budapest határa és a Duna vonala látható!

8.7 gyakorlat: Készítsünk programot, amelyik a logaritmusfüggvény grafikonját rajzolja ki koordináta-rendszerben!

8.8 gyakorlat: Módosítsuk a 8.4 feladat megoldását úgy, hogy általunk megadott kezdőértéktől, kívánt lépésközzel íródjanak ki egy billentyű lenyomására a táblázat sorai!

8.9 gyakorlat: Jelenjen meg négy négyzet a képernyőn, majd véletlenszerűen ugráljon egyikből a másikba egy felkiáltójel!

8.10 gyakorlat: A képernyőn megjelenő 9 × 9-es sakktáblán véletlenszerűen lépkedjen egy bábu! A bábu grafikus pontokból álljon.

8.11 gyakorlat: Készítsünk stoppert a képernyőre! Az [I] billentyű lenyomására induljon, az [S] billentyű lenyomására álljon meg, a [Ø] billentyű lenyomására pedig nullázódjon.

8.12 gyakorlat: Készítsünk olyan programot, amelyik kiír 16 darab, DATA utasításokkal tárolt számot egymás alá, majd mindegyik mellé a négyzetét, és végül ezek mellé a köbét!

8.13 gyakorlat: Jelenjen meg 5Ø darab véletlenszerűen generált karakter a képernyőn egymás mellett, majd fordított sorrendben tűnjön el!

8.14 gyakorlat: Rajzoltassunk véletlenszerűen kiválasztott karakterből egy betűt a képernyőre! Egy billentyű lenyomására ezt ismételje a program.

8.15 gyakorlat: Rajzoltassunk véletlenszerűen kiválasztott karakterekből egy betűt a képernyőre! Egy billentyű lenyomására ezt ismételje a program. (Ez a gyakorlat különbözik a 8.14 gyakorlattól!!!)

8.16 gyakorlat: Készítsünk olyan programot, amelyik kiír a képernyő közepére egy véletlenszámot véletlenül kiválasztott karakterekből álló keretben!

8.17 gyakorlat: Generálja a számítógép véletlenszerűen a 128-tól 191-ig terjedő ASCII-kódú karaktereket, és közben rakja azokat a helyükre egy 8 × 8-as táblázatba úgy, hogy annak a bal felső helyére a 128-as, mellé a 129-es ... stb. ASCII-kódú karakter kerüljön!

8.18 gyakorlat: Rajzoltassunk egyszerű figurát a képernyőre grafikus karakterekből!

8.19 gyakorlat: Készítsünk olyan programot, amelynek ha megadunk egy -val, -vel ragos szót, amelynek szótöve sz-re végződik, akkor kiírja a szót valamely sor végére elválasztva!

A nyolcadik fejezet feladatainak egy lehetséges megoldása

8.1 feladat

Ha az I értékének egészrésze negatív vagy 1Ø23-nál nagyobb, akkor hibajelzést kapunk, egyébként pedig az adott I értékének egészrészével dolgozik a program.

8.2 feladat

```
10 DEFINT I,J
20 CLS
30 REM KERETRAJZOLAS
40 FOR I=0 TO 63 : PRINT@I,"*"; : PRINT@14*64+I,"*"; : NEXT I
50 FOR J=1 TO 14 : PRINT@J*64,"*"; : PRINT@J*64+63,"*"; : NEXT J
60 REM SZOVEGIRAS A KERETBE
70 PRINT@414,"N E V"; : PRINT@542,"C I M";
80 GOTO 80
```

8.3 feladat

A bevezetőben megtalálható ennek a feladatnak a megoldása!

8.4 feladat

```
10 DEFINT I
20 CLS
30 X=0
40 PRINT" X";TAB(14);"INT(X)";TAB(27);"25*X";TAB(40);"X*X";
   TAB(53);"SQR(X)"
50 FOR I=0 TO 63 : PRINT "-"; : NEXT I
60 FOR I=2 TO 15
70   PRINT@I*64,X;TAB(13);INT(X);TAB(26);25*X;TAB(39);X*X;
     TAB(52);SQR(X);
80   X=X+.5
90 NEXT I
100 GOTO 100
```

A 7Ø-es sorban azért van kurzor címzés, hogy közvetlenül a gondolatjelekből álló vonal alatt kezdődhessen a kiíratás.

8.5 feladat

```
10 DEFINT I
20 CLS
30 READ I
40 IF I=1024 THEN 70
50 READ A$
60 PRINT@I,A$; : GOTO 30
70 GOTO 70
80 REM AZ ALAKZAT ADATAI
90 DATA 286,*****,349,* A A *,413,*  T  *,477,*  V  *,542,*****,1024
```

8.6 feladat

```
10 DEFINT I
20 CLS
30 READ I
40 IF I=1024 THEN 100 ELSE PRINT@I,"";
50 READ A$
60 IF A$="&" THEN 30
70 FOR I1=0 TO 100 : NEXT
80 PRINT A$;
90 GOTO 50
100 GOTO 100
110 REM A KIIRANDO SZOVEG
120 DATA 279,K,E,D,V,E,S," ",B,A,R,A,T,U,N,K," ",!,&,525,
    T,O,V,A,B,B,I," ",L,E,L,K,E,S," ",P,R,O,G,R,A,M,O,Z,A,S,T,
    " ",K,I,V,A,N,U,N,K," ",!,!,!,&,1024
```

8.7 feladat

```
10 DEFINT I
20 CLS
30 REM AZ ABRA MEGRAJZOLASA
40 READ I1
50 IF I1=1024 THEN 70
60 READ I2 : SET(I1,I2) : GOTO 40
70 REM FELIRATOK
80 READ I
90 IF I=1024 THEN 110
100 READ A$ : PRINT@I,A$; : GOTO 80
110 GOTO 110
120 REM AZ ABRA ADATAI
130 DATA 60,15,58,14,56,13,55,12,55,10,56,7,58,6,60,6,62,7,63,
         9,64,10,65,11,64,12,63,13,62,14
140 DATA 58,16,56,15,54,15,51,15,48,16,43,19,44,21,49,21,51,20,
         53,19,55,18,56,17,58,16
150 DATA 63,15,65,15,68,15,72,16,76,17,78,18,79,19,77,20,75,21,
         72,21,69,20,65,18,64,17,62,16
160 DATA 60,17,62,19,62,21,60,23,58,25,55,27,52,29,49,31,47,33,
         45,35,43,37,42,40,41,43,40,47,1024
170 DATA 65,EZ EGY VIRAG,857,<---- EZ A ...,1024
```

8.8 feladat

```
10 DEFINT I
20 CLS
30 REM A FEJ RAJZOLASA
40 PRINT@285," *****"; : PRINT@285+4*64," *****"; :
   PRINT@285+64,"* A A *"; : PRINT@285+128,"*  T  *"; :
   PRINT@285+192,"*  V  *";
50 FOR I=0 TO 300 : NEXT
60 PRINT@285+68,"-"; : PRINT@288+192,"O";
70 FOR I=0 TO 300 : NEXT
80 PRINT@285+68,"A"; : PRINT@288+192,"V";
90 GOTO 50
```

8.9 feladat

A 3Ø-as sorba pontosvessző helyett vessző kell. Ha a PRINT utasítás listájának a végére pontosvesszőt írunk, akkor az a kiíratás után nem emel sort.

8.10 feladat

Az Y változót karaktertípusúként kell definiálni. A DATA listájában a szükségesnél eggyel kevesebb elem áll.

Érdemes megvizsgálni azt, hogy mi történik akkor, ha a 3Ø-as sorban a PRINT kulcsszó után a „@" jelet is kitesszük...

8.11 feladat

A program szintaktikus hibát nem tartalmaz, futásának eredménye mégsem az lesz, mint amit az első ránézésre várnánk. Azért, mert ha a PRINT utasítás segítségével az 1Ø23. karakterpozícióba írunk, akkor a számítógép mindenképpen végrehajt egy soremelést.

Kilencedik fejezet

MAGASABB DIMENZIÓK

Tömbök

Módosítsuk az 1.6 feladatot úgy, hogy nem 4, hanem 24 pont koordinátáit kell megadni, majd ezután kigyújtani ezeket a pontokat!

Eddigi ismereteink birtokában természetesen megoldható ez a feladat is, de bizonyára nem lelkesedünk érte túlságosan, hiszen $2 \cdot 24 = 48$ darab változó használatára lenne szükség. Használhatunk azonban ún. *indexes változó*kat vagy más néven *tömb*öket, és ezek segítségével már egyszerűen megoldható a probléma. Definiálhatunk a – $2 \cdot 24$ változó helyett – 2 darab, egyenként 24 elemből álló tömböt, és a pontok koordinátáit ezekben tárolhatjuk. Ezek valamely elemére az **X(J)**, illetve az **Y(J)** formában hivatkozhatunk. A tömbök indexezése nullával kezdődik, tehát a 24 elemű X tömb elemei: X(Ø), X(1), X(2), ..., X(22), X(23). Ennek megfelelően az X tömb J-edik eleme az X(J), ahol $Ø \leqq J \leqq 23$. A tömbelemekkel ugyanúgy dolgozhatunk mint a változókkal, és azokhoz hasonlóan a tömböket is definiálhatjuk egész típusúaknak, valós típusúaknak, duplapontosságúaknak vagy karaktertípusúaknak.

A tömbök méreteit a **DIM** utasítással definiálhatjuk.

9.1 példaprogram

```
10 REM AZ 1.6 FELADAT MODOSITASA
20 DEFINT I,X,Y
30 CLS
40 DIM X(23),Y(23)
50 PRINT"KEREM A 24 PONT KOORDINATAIT !"
60 FOR I=0 TO 23 : INPUT X(I),Y(I) : NEXT I
70 CLS
80 FOR I=0 TO 23 : SET(X(I),Y(I)) : NEXT I
90 GOTO 90
```

Megjegyzés: Ha egy tömb méreteit nem definiáljuk, akkor a számítógép automatikusan 11 eleműnek veszi, tehát indexének lehetséges legnagyobb értéke 1Ø lehet.

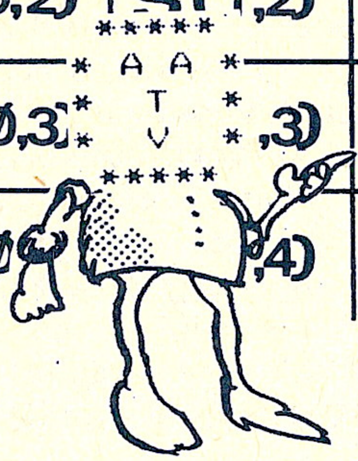
9.1 feladat: Módosítsuk a 9.1 példaprogramot úgy, hogy 24 különböző pont koordinátáit kérje be a számítógép, majd ezután gyújtsa is ki őket!

9.2 feladat: Véletlenszerűen jelenjenek meg a hónapok nevei a képernyő közepén!

9.3 feladat: A 6.9 feladatra adott megoldásunkat módosítsuk úgy, hogy az oszlopok növekedése ne lassuljon, a számok kiíratásához pedig használjuk a PRINT@ utasítást!

A 9.1 példaprogramban használt X és Y tömbök egydimenziósak voltak. Használhatunk azonban két- vagy többdimenziós tömböket is. A kétdimenziós (például 4 oszlopból és 5 sorból álló 4 × 5-ös) S tömb elemeire az S(I,J) formában hivatkozhatunk. Az S(2,3) tehát a tömb 2. oszlopának 3. eleme. A sorok és az oszlopok számozása itt is nullával kezdődik!

	0. OSZLOP	1. OSZLOP	2. OSZLOP	3. OSZLOP
0. sor	S(0,0)	S(1,0)	S(2,0)	S(3,0)
1. sor	S(0,1)	S(1,1)	S(2,1)	S(3,1)
2. sor	S(0,2)	S(1,2)	S(2,2)	S(3,2)
3. sor	S(0,3	,3)	**S(2,3)**	S(3,3)
4. sor	S(0	,4)	S(2,4)	S(3,4)



A következő példaprogramunkban a 128 × 48-as K tömböt feltöltjük 0 és 101 közé eső véletlen egész számokkal. A tömb K(X,Y) elemét feleltessük meg a képernyő (X,Y) koordinátájú pontjának, és pontosan akkor gyújtsuk ki azt, ha a neki megfelelő tömbelem értéke öttel osztható:

9.2 példaprogram

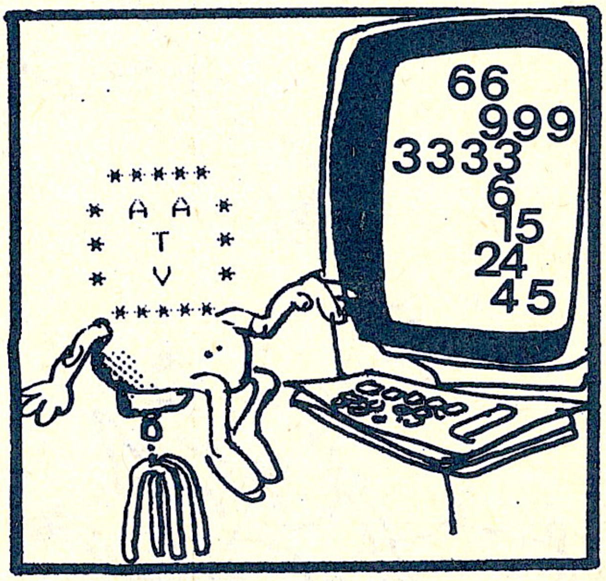
```
10 DEFINT K,X,Y : DIM K(127,47)
20 CLS :
   PRINT@6*64+5,"FELTOLTOM A TOMBOT !"
30 FOR X=0 TO 127 : FOR Y=0 TO 47 : K(X,Y)=RND(100) : NEXT Y,X
40 CLS
50 FOR X=0 TO 127
60   FOR Y=0 TO 47
70     K=K(X,Y)/5
80     IF 5*K=K(X,Y) THEN SET(X,Y)
90 NEXT Y,X
100 GOTO 100
```

9.4 feladat: Írassuk ki egy 9 × 9-es tömb elemeit a képernyőre többféle elrendezésben is! (A tömb elemeit előzetesen töltsük fel véletlenszámokkal!)

Általában akárhány dimenziós tömböket is használhatunk. A dimenziószámnak egyedül az szab határt, hogy a tömb elférjen a memóriában.

Megjegyzés: Ha egy többdimenziós tömb egyik indexének értéke sem lehet nagyobb 10-nél, akkor ezt a tömböt nem kell a DIM utasítással definiálni. Ezt általában mégis tanácsos megtenni, mert így a tömb számára kevesebb helyet foglalunk le a memóriában.

A DIM utasításban szereplő tömbök indexei nemcsak konkrét számok, hanem változók, illetve aritmetikai kifejezések is lehetnek. Ezek aktuális értékének nemnegatív számoknak kell lenniük akkor, amikor a DIM utasítás végrehajtásra kerül. Ezt nevezik *dinamikus tömbdefiniálás*nak.



9.5 feladat: Készítsünk olyan programot, amelyik beolvas *N* darab számot, majd ezek közül kiírja a hárommal oszthatókat! Az *N* bemenő adat legyen.

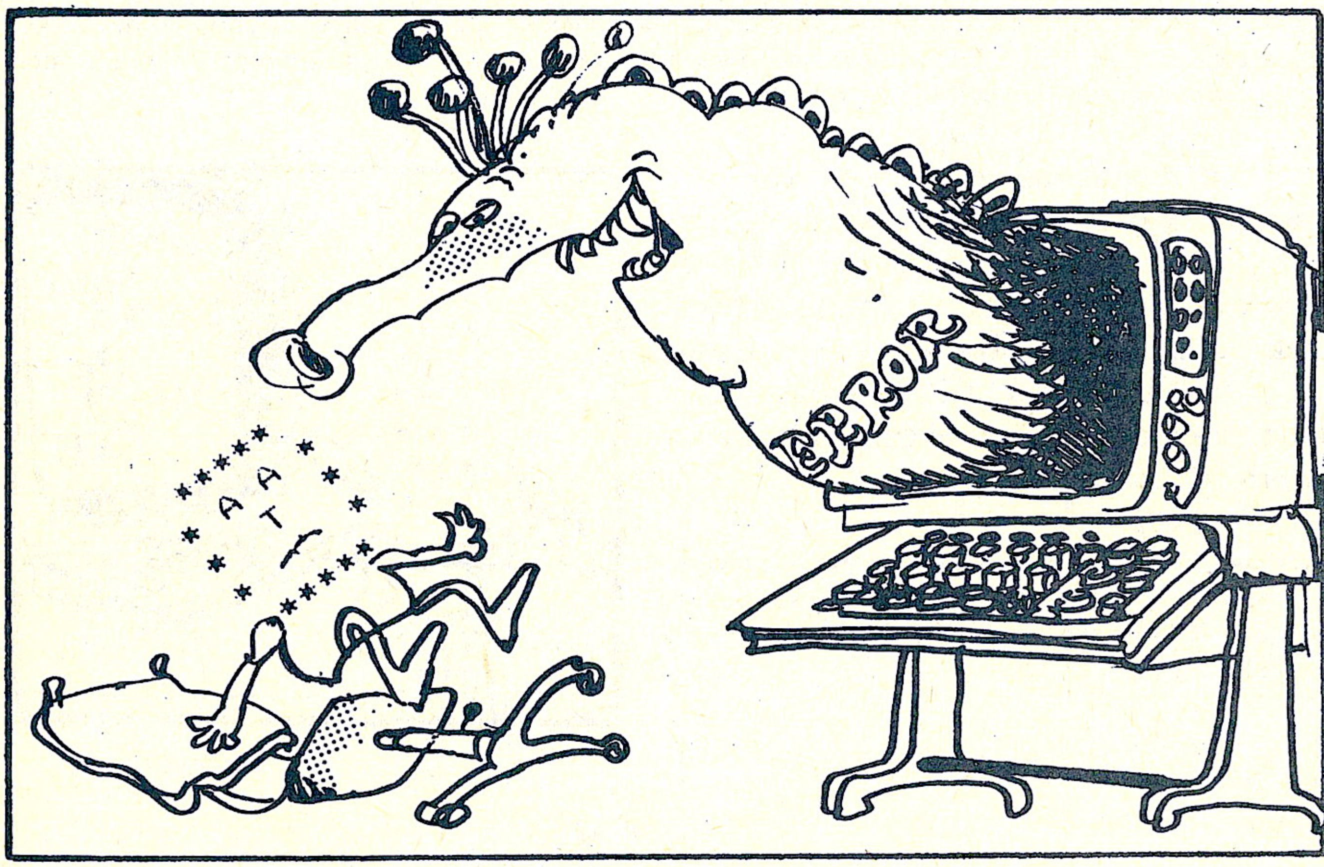
CLEAR

Tanulmányozzuk át a következő példaprogramot!

9.3 példaprogram

```
10 DEFINT I,N,X
20 CLS
30 INPUT"HANY DARAB SZAM LESZ ";N : N=N-1
40 DIM X(N)
50 FOR I=0 TO N : INPUT X(I) : NEXT I
60 CLS
70 PRINT@3*64,"A HAROMMAL OSZTHATOK :" : PRINT
80 FOR I=0 TO N :
     X=X(I)/3 :
     IF X(I)=3*X THEN PRINT X(I),
90 NEXT I
100 PRINT@14*64,"HA FOLYTATOD, NYOMD LE A T-BILLENTYUT !";
110 A$=INKEY$
120 IF A$="T" THEN 130 ELSE 110
130 CLEAR
140 GOTO 10
```

Észrevehettük, hogy ez a program megoldása lehet a 9.5 feladatnak, de tartalmaz egy olyan utasítást, amellyel már a hetedik fejezetben megismerkedtünk, ezúttal azonban új szerepe van. Próbáljuk ki, hogy mi történik, ha a 13Ø-as sorból elhagyjuk a CLEAR utasítást, és úgy futtatjuk le a programot!



A hibajelzés oka az volt, hogy amikor a program másodszor érkezett a tömböt definiáló DIM X(N) utasításhoz, a számítógép az egyszer már definiált X tömböt akarta újradimenzionálni. Ez csak úgy lehet, ha előtte a CLEAR utasítás segítségével a tömb korábban meghatározott méretét töröljük. (Ne feledjük továbbá azt sem el, amit már a hetedik fejezetben láttunk, hogy a CLEAR utasítás ezen kívül még törli az összes változó típusát és értékét is!)

9.6 feladat: Hol a hiba?

```
1Ø CLS
2Ø PRINT@6*64,"A 13-NAL KISEBB POZITIV EGESZEK :"
3Ø FOR I=1 TO 12
4Ø   A(I)=I
5Ø NEXT I
6Ø FOR J=12 TO 1 STEP -1
7Ø   PRINT A(J),
8Ø NEXT J
```

9.7 feladat: Hol a hiba?

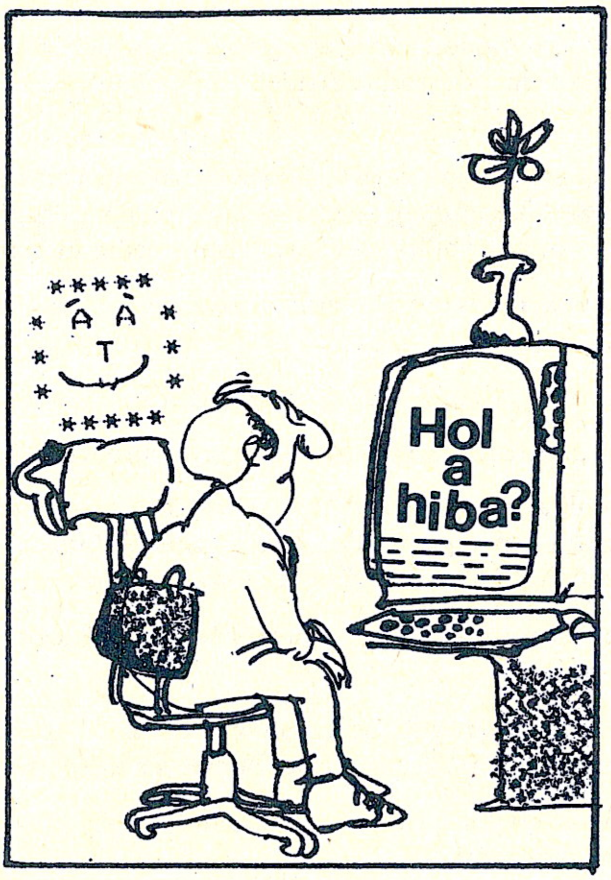
```
10 FOR I=1 TO 5 :
      FOR J=1 TO 5 :
         FOR K=1 TO 5 :
            FOR L=1 TO 5 :
               B(I,J,K,L)=RND(10) :
   NEXT L,K,J,I
```

9.8 feladat: Hol a hiba?

```
20 CLS
30 INPUT N : DIM A(N)
40 FOR J=1 TO N
50    A(J)=J*J : PRINT SQR(A(J))/A(0)
60 NEXT
70 FOR K=1 TO 300 : NEXT
80 GOTO20
```

9.9 feladat: Hol a hiba?

```
10 VELETLEN=RND(20) :DIM TOMB(VELETLEN) : OT=5
20 CLS
30 INPUT EGESZ%,FEL,DUPLAPRECIZ#,STRINGECSKE$
40 CLS
50 MENTO%=EGESZ%/16 : EGESZ%=ABS(EGESZ%-MENTO%*16)
60 ALLJ=VELETLEN : INDULJ=INT(SQR(VELETLEN)/OT)
70 FOR ROHANJ=INDULJ TO ALLJ
80    TOMB(ROHANJ)=FEL*DUPLAPRECIZ#-ROHANJ
90    PRINT@EGESZ%*64+VELETLEN,CHR$(ASC(STRINGECSKE$));
      TOMB(ROHANJ)
100 NEXT ROHANJ
110 REMELEM,HOGY TETSZETT !
```



Gyakorlatok a kilencedik fejezet anyagához

9.1 gyakorlat: Írassuk ki a képernyőre a "VELETLEN" feliratot, de a betűk véletlenszerű sorrendben jelenjenek meg!

9.2 gyakorlat: Készítsünk programot, amelyik bekér 1Ø nevet, majd véletlen párosításban kiírja azokat!

9.3 feladat: Jelenjen meg 2Ø véletlenszám a képernyőn a generálás sorrendjében, majd ugyanezek, de csökkenő sorrendben!

9.4 gyakorlat: Készítsünk olyan programot, amelyik kiírja a PASCAL-háromszög első 1Ø sorát!

9.5 gyakorlat: Egy tanuló ellenőrzőjében 1Ø tantárgy számára tantárgyanként 6 osztályzat beírására van lehetőség egy hónapban. Készítsünk programot, amelynek segítségével az ellenőrzőt helyettesíthetjük, jegyeket írhatunk be, illetve meg is nézhetjük, hogy valamely tantárgyból milyen jegyek vannak beírva!

9.6 gyakorlat: Végeztessünk a számítógéppel 1ØØØ darab lottószám húzást, majd ezután kérésre mondja meg a program, hogy valamely számot hányszor húzott ki!

9.7 gyakorlat: Hajtassuk végre 1ØØØ-szer a számítógéppel az RND(5Ø) függvényutasítást, és grafikusan ábrázoljuk a kapott véletlenszámok eloszlását!

9.8 gyakorlat: Készítsünk programot, amelyik egy adott számig kiírja a prímszámokat!

9.9 gyakorlat: Készítsünk programot, amelyik beolvas 16 darab számot, majd kiírja az ezeknek megfelelő ASCII-kódú karaktereket, ha ilyenek léteznek!

9.10 gyakorlat: Készítsünk programot, amelyik egymás után bekéri egy szó betűit, majd kiírja azokat ASCII-kódok szerint sorrendbe rendezve egy újabb szó formájában! A szó hossza bemenő adat legyen.

9.11 gyakorlat: Írjunk olyan programot, amelyik 2Ø véletlenszerűen kiválasztott betűből véletlenszerűen kiválaszt ötöt, ezeket összefűzi egy szóvá, majd kiírja a képernyőre! Ezután az [U] billentyű lenyomására új szót ír ki, a [B] billentyű lenyomására pedig új betűkészlettel kezdi elölről.

9.12 gyakorlat: Rajzoltassunk öt darab *A* betűt a képernyőre grafikus (128–191 ASCII-kódú) karakterekből!

9.13 gyakorlat: Készítsünk programot a MEMÓRIA-játékra!

9.14 gyakorlat: Készítsünk programot a TORPEDÓ (vagy más néven TENGERI CSATA)-játékra!

9.15 gyakorlat: Végeztessünk a számítógéppel 1ØØØ darab lottószám húzást! Ezután jelenjen meg a képernyőn vagy számszerűen vagy grafikusan szemléltetve (attól függően, hogy mit kértünk) az, hogy az egyes számokat hányszor húzta ki a számítógép.

9.16 gyakorlat: Készítsünk olyan programot, amelyik beolvas 1Ø szót, majd hosszúságuk szerint sorrendben kiírja őket!

9.17 gyakorlat: Készítsünk olyan programot, amelyik beolvas 16 egész számot, majd úgy írja ki egymás alá őket, hogy az azonos helyiértékű számjegyek egymás alá kerüljenek!

9.18 gyakorlat: Készítsünk nyelvtani gyakorló programot, amelyik a *j*-t vagy az *ly*-t tartalmazó szavak helyesírását gyakoroltatja!

A kilencedik fejezet feladatainak egy lehetséges megoldása

9.1 feladat

```
10 CLS
20 DEFINT I,J,X,Y
30 DIM X(23),Y(23)
40 PRINT"KEREM A 24 PONT KOORDINATAIT !"
50 FOR I=0 TO 23
60   INPUT X(I),Y(I)
70   IF I=0 THEN 110
80   FOR J=0 TO I-1
90     IF X(I)=X(J) AND Y(I)=Y(J) THEN
       PRINT"ILYEN MAR VOLT !" : GOTO 60
100  NEXT J
110 NEXT I
120 CLS
130 FOR I=0 TO 23 : SET(X(I),Y(I)) : NEXT I
140 GOTO 140
```

9.2 feladat

```
10 DEFSTR A : DIM A(11)
20  A(0)="JANUAR" : A(1)="FEBRUAR" : A(2)="MARCIUS" :
    A(3)="APRILIS" : A(4)="MAJUS" : A(5)="JUNIUS" :
    A(6)="JULIUS" : A(7)="AUGUSZTUS" : A(8)="SZEPTEMBER" :
    A(9)="OKTOBER" : A(10)="NOVEMBER" : A(11)="DECEMBER"
30 CLS
40 PRINT@6*64+20,A(RND(12)-1)
50 FOR I=1 TO 300 : NEXT
60 GOTO 30
```

9.3 feladat

```
10 CLS
20 DEFINT I,X,Y
30 FOR I=1 TO 10 : PRINT@5+I*5,I : NEXT I
40 I=RND(10) : X(I)=X(I)+1 : X=12+10*I : Y=47-X(I)
50 IF Y>2 THEN SET(X,Y) : GOTO 40
60 GOTO 60
```

A programból látható, hogy a számítógép megkülönbözteti az X változót és az X tömböt.

9.4 feladat

```
10 DIM A(9,9)
20 FOR I=0 TO 9 : FOR J=0 TO 9 : A(I,J)=RND(9) : NEXT J,I
30 CLS
40 FOR I=0 TO 9 : FOR J=0 TO 9 : PRINT A(I,J); : NEXT J,I
50 FOR I=0 TO 9 : FOR J=0 TO 9 : PRINT@(I+6)*64+2*J,A(I,J); : NEXT J,I
60 FOR J=0 TO 9 : FOR I=0 TO 9 : PRINT@(I+6)*64+30+2*J,A(J,I); : NEXT I,J
70 GOTO 70
```

9.5 feladat

```
10 DEFINT I,N,X
20 CLS
30 INPUT"HANY DARAB SZAM LESZ ";N : N=N-1
40 DIM X(N)
50 FOR I=0 TO N : INPUT X(I) : NEXT I
60 CLS
70 PRINT@3*64,"A HAROMMAL OSZTHATOK :" : PRINT
80 FOR I=0 TO N
90   X=X(I)/3
100  IF X(I)=3*X THEN PRINTX(I),
110 NEXT I
120 STOP
```

9.6 feladat

A 11-nél nagyobb elemszámú tömböket definiálni kell.

9.7 feladat

Mivel a négydimenziós *B* tömb méreteit nem definiáltuk, ezért a számítógép azt automatikusan egy

$$11 \times 11 \times 11 \times 11\text{-es}$$

tömbként definiálja. Ekkora tömb a memóriában nem fér el, ezért kapunk hibajelzést. Ha a *B* tömböt a

DIM *B*(5,5,5,5)

utasítással a program elején definiáljuk, akkor az már működőképes lesz.

9.8 feladat

Az *A*(Ø) tömbelem értéke Ø (amennyiben korábban nem adtunk neki más értéket), ezért az 5Ø-es sorban hibajelzést kapunk.

Ha az *A* tömb méretét egyszer már definiáltuk egy programban, akkor újradefiniálás előtt a korábbi méretet egy CLEAR utasítással törölni kell. Ezt az utasítást beírhatjuk például a 25-ös sorba.

9.9 feladat

A változónevek nem tartalmazhatnak BASIC-kulcsszavakat!

(VELETLEN – LET, LEN
STRINGECSKE – STR, STRING
TOMB – TO
MENTO – TO)

Tizedik fejezet

EGY PROGRAM PROGRAMOKAT IS TARTALMAZHAT

Egy program megtervezése

Írjunk együtt programot a következő feladatra:
Mozogjon véletlenszerűen egy bábu a képernyőn megjelenő 8 × 8-as sakktáblán!

Ez a feladat alapvetően két kis program kidolgozását kívánja:
1. Sakktáblarajzoló
2. Bábumozgató

Vizsgáljuk meg egyes részleteiket közelebbről!
1. A sakktáblarajzoló programot még további részekre bonthatjuk:
Rajzolást vezérlő
Négyzetrajzoló

Ehhez tervezzük meg a sakktábla méreteit! Egy mezője – a grafikus pont alakját figyelembe véve – legyen 1Ø × 5-ös. Ennek megfelelően ezek bal felső pontjainak koordinátái (A, B):

(Ø,Ø)	(1Ø,Ø)	...	(7Ø,Ø)
(Ø,5)	(1Ø,5)	...	(7Ø,5)
.	.	...	.
.	.	...	.
(Ø,35)	(1Ø,35)	...	(7Ø,35)

Az egyes mezőket ezekkel a koordinátákkal fogjuk jellemezni.

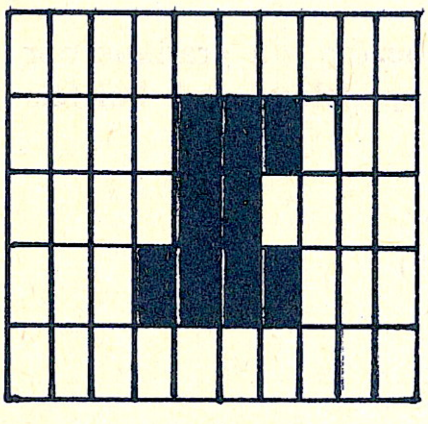
Pontosan azokat a részeket kell fehérre festeni, amelyek első koordinátájának tizedrésze és a második koordinátája egyszerre páros vagy egyszerre páratlan. Ezt felhasználva a sakktáblarajzoló program a következő:

10.1 példaprogram

```
10 REM SAKKTABLAT  RAJZOLO PROGRAM
20 CLS
30 DEFINT A-Z
40 REM A RAJZOLAST VEZERLO RESZ
50 FOR A=0 TO 70 STEP 10
60   FOR B=0 TO 35 STEP 5
70     X=A/20 : Y=B/2
80     IF A=20*X AND B=2*Y THEN 100
90     IF A>20*X AND B>2*Y THEN 100 ELSE 150
100    REM NEGYZETRAJZOLO
110    FOR I=0 TO 9
120      FOR K=0 TO 4
130        SET(A+I,B+K)
140    NEXT K,I
150 NEXT B,A
160 GOTO 160
```

2. A bábumozgató programot is két részre bonthatjuk:
   Mozgatást vezérlő
   Báburajzoló

Először a 10 × 5-ös mezőben egy 9 grafikus pontból álló bábut tervezzünk. Ennek a bábunak a koordinátái az (A,B) mezőben:



(A+4,B+1), (A+5,B+1), (A+6,B+1),
(A+4,B+2), (A+5,B+2),
(A+3,B+3), (A+4,B+3), (A+5,B+3), (A+6,B+3)

Készítsük el a programot úgy, hogy fehér mezőben ezeket a pontokat kikapcsoljuk, feketében pedig kigyújtjuk.

10.2 példaprogram

```
10 REM BABURAJZOLO
20 RESTORE
30 FOR I=1 TO 9
40   READ X,Y
50   IF POINT(A+X,B+Y) THEN RESET(A+X,B+Y) ELSE SET(A+X,B+Y)
60 NEXT I
70 GOTO 70
80 DATA 4,1,5,1,6,1,4,2,5,2,3,3,4,3,5,3,6,3
```

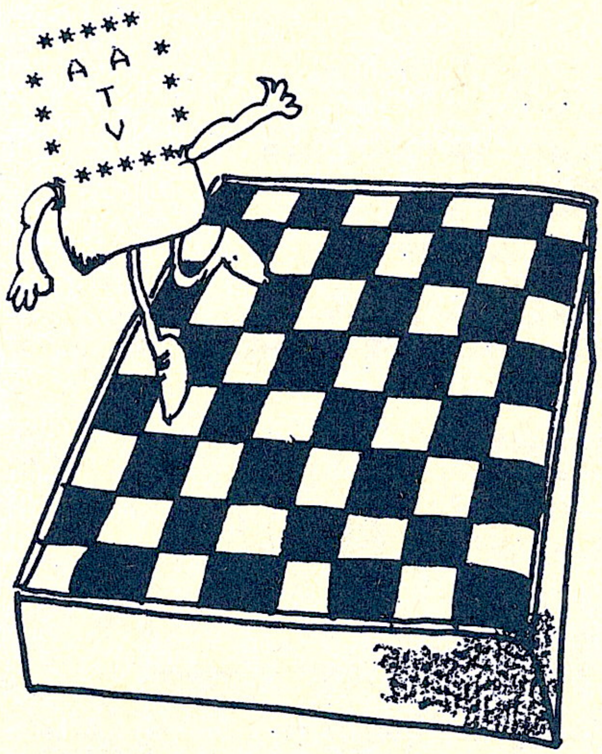
Vegyük észre, hogy ezt a programot ugyanazon értékekre kétszer egymás után lefuttatva kirajzolja a bábut, és utána le is törli!

A bábumozgatás vezérlésének megírása már nem nehéz feladat, így a bábumozgató program következő:

10.3 példaprogram

```
10 REM BABUMOZGATO
20 A=(RND(8)-1)*10 : B=(RND(8)-1)*5
30 REM BABURAJZOLO
40 RESTORE
50 FOR I=1 TO 9
60   READ X,Y
70   IF POINT(A+X,B+Y) THEN RESET(A+X,B+Y) ELSE SET(A+X,B+Y)
80 NEXT I
90 FOR I=1 TO 200 : NEXT
100 REM BABURAJZOLO
110 RESTORE
120 FOR I=1 TO 9
130   READ X,Y
140   IF POINT(A+X,B+Y) THEN RESET(A+X,B+Y) ELSE SET(A+X,B+Y)
150 NEXT I
160 GOTO 20
170 DATA 4,1,5,1,6,1,4,2,5,2,3,3,4,3,5,3,6,3
```

10.1 feladat: A fejezetek eddigi példaprogramjainak felhasználásával készítsünk olyan programot, amelyik véletlenszerűen mozgat a képernyőn megjelenő 8 × 8-as sakktáblán egy bábut!



Szubrutinok

Az előző problémára megoldásként adott programban előfordul olyan programrész (a báburajzoló), amit többször is leírtunk. Jó lenne, ha ezt csak egyszer kellene elvégezni, és amikor szükség van rá, mindig csak hivatkoznánk az egyszer leírt programrészre.

Az is látható, hogy ezt a programot több kisebb, már előre elkészített részből állítottuk össze. Az előbbi gondunk egyszerűen megoldható, és az ilyen típusú programszerkesztés lényegesen megkönnyíthető a szubrutinok felhasználásával.

A *szubrutin*ok bizonyos értelemben önálló programrészek, alprogramok, amelyekre mint BASIC-programokra, minden eddig megismert szabály érvényes. A szubrutint a főprogramban elhelyezett **GOSUB sorszám** utasítással hívhatjuk meg, ahol a sorszám a szubrutin kezdő sorának sorszáma. A szubrutinnak tartalmaznia kell legalább egy **RETURN** utasítást, amelynek végrehajtásakor a vezérlés a szubrutint meghívó GOSUB utasítást követő utasításra adódik át.

A 10.1 feladatot szubrutinok felhasználásával a következőképpen lehet megoldani:

10.4 példaprogram

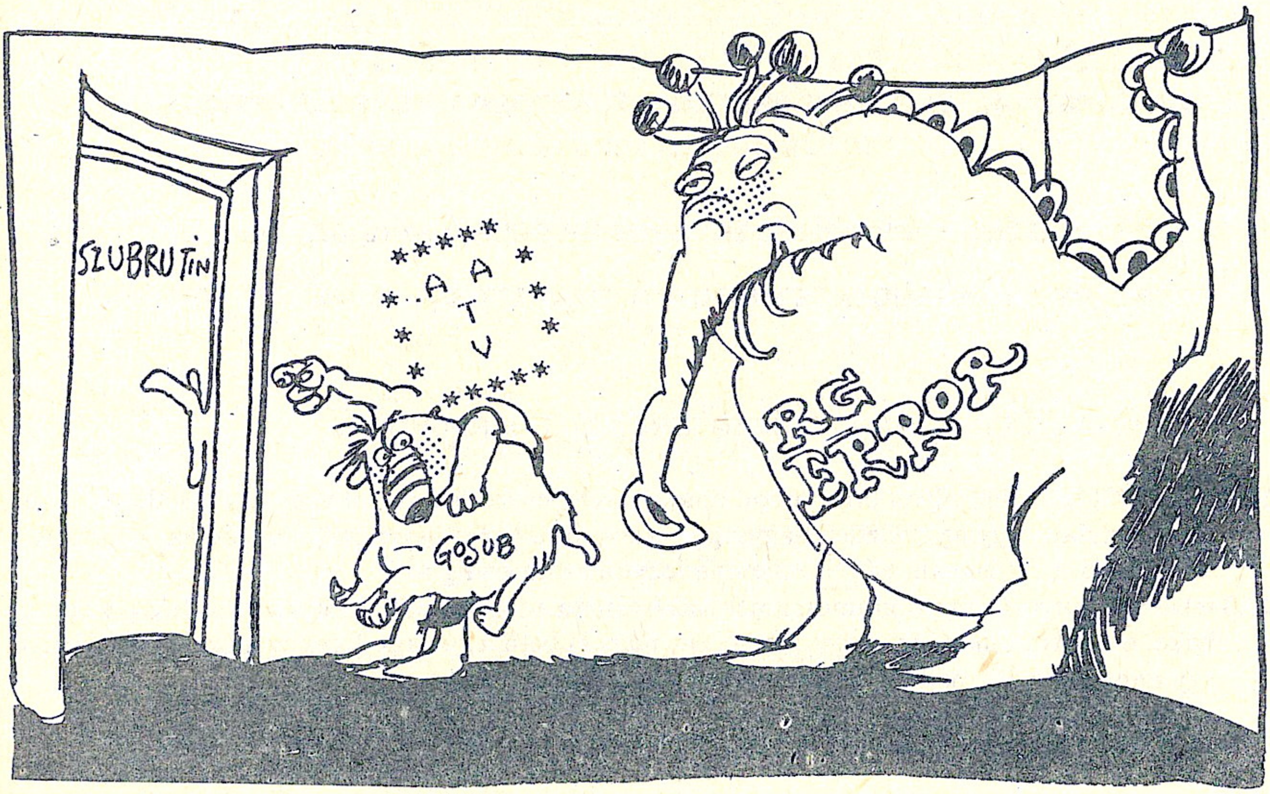
```
10 REM SAKKTABLAN BABUT MOZGATO PROGRAM
20 CLS
30 DEFINT A-Z
40 GOSUB 70
50 GOSUB 220
60 GOTO 50
70 REM SAKKTABLAT  RAJZOLO RESZ
80 REM RAJZOLAST VEZERLO RESZ
90 FOR A=0 TO 70 STEP 10
100   FOR B=0 TO 35 STEP 5
110     X=A/20 : Y=B/2
120     IF A=20*X AND B=2*Y THEN GOSUB 160
130     IF A>20*X AND B>2*Y THEN GOSUB 160
140 NEXT B,A
150 RETURN
160 REM NEGYZETRAJZOLO
170 FOR I=0 TO 9
180   FOR K=0 TO 4
190     SET(A+I,B+K)
200 NEXT K,I
210 RETURN
220 REM BABUMOZGATO
230 A=(RND(8)-1)*10 : B=(RND(8)-1)*5
240 GOSUB 280
250 FOR I=1 TO 200 : NEXT
260 GOSUB 280
270 RETURN
```

```
28Ø REM BABURAJZOLO
29Ø RESTORE
3ØØ FOR I=1 TO 9
31Ø   READ X,Y
32Ø   IF POINT(A+X,B+Y) THEN RESET(A+X,B+Y) ELSE SET(A+X,B+Y)
33Ø NEXT I
34Ø RETURN
35Ø REM A BABU ADATAI
36Ø DATA 4,1,5,1,6,1,4,2,5,2,3,3,4,3,5,3,6,3
```

A 1Ø–6Ø-as sorokat tekinthetjük egy főprogramnak. Ez meghívja a
SAKKTABLAT RAJZOLO RESZ
elnevezésű szubrutint, ami megrajzolja a sakktáblát, majd állandóan ismételve meghívja a
BABUMOZGATO
szubrutint, ami a bábu rajzolását és mozgatását végzi. A sakktáblarajzoló szubrutin többször meghívja a
NEGYZETRAJZOLO
szubrutint, a bábumozgató rész pedig a
BABURAJZOLO
szubrutint.

Megjegyzés: Az utolsó mondatunkban leírtakat szubrutinok egymásba skatulyázásának nevezzük.

Érdemes megvizsgálni, hogy mi történik akkor, ha a 10.4 példaprogramból a 6Ø-as sort töröljük, és a programot lefuttatjuk!



A hibaüzenet oka az, hogy a főprogramból a GOSUB utasítás nélkül jutottunk el a 17Ø-es sorban levő RETURN utasításhoz. Ennek az elkerülésére példát láthatunk a következő programvázlatban, aminek tanulmányozása hasznos a szubrutinok alkalmazásához is:

10.5 példaprogram

```
10 REM PROGRAM SZUBRUTINOKKAL
20 REM A FOPROGRAM A 10-600 SOROKBAN
   .
   .
   .
80 REM AZ ELSO SZUBRUTIN MEGHIVASA KOVETKEZIK
90 GOSUB 2000
100 AZ ELSO SZUBRUTIN VEGREHAJTASA UTAN IDE TERUNK VISSZA
   .
   .
   .
150 REM UJRA AZ ELSO SZUBRUTIN MEGHIVASA KOVETKEZIK
160 GOSUB 2000 : REM IDE TERUNK VISSZA
   .
   .
   .
230 REM A MASODIK SZUBRUTIN MEGHIVASA KOVETKEZIK
240 GOSUB 2500
250 REM A MASODIK SZUBRUTIN VEGREHAJTASA UTAN IDE TERUNK VISSZA
   .
   .
   .
600 END : REM A PROGRAM VEGE
   .
   .
   .
2000 REM AZ ELSO SZUBRUTIN UTASITASAI KOVETKEZNEK
   .
   .
   .
2320 RETURN : REM VISSZATERES A FOPROGRAMHOZ

2500 REM A MASODIK SZUBRUTIN UTASITASAI KOVETKEZNEK
   .
   .
   .
2700 RETURN : REM VISSZATERES A FOPROGRAMHOZ
```

10.2 feladat: Készítsünk olyan programot, amelyik véletlenszerűen rajzol vagy egy véletlen hosszúságú vonalat, vagy egy véletlen nagyságú téglalapot a képernyőre, majd ezután az [I] billentyű lenyomására ezt ismétli, az [N] billentyű lenyomására pedig leáll!

10.3 feladat: Írjunk olyan programot, amelynek bemenő adatként a T, H, TH, HT értékek valamelyikét megadva rendre egy nagy T betű, egy nagy H betű, egy nagy T és egy nagy H betű, illetve egy nagy H és egy nagy T betű rajzolódik ki a képernyőre!

Érdemes a korábbi fejezetek gyakorlatai és feladatai közül megkeresni azokat, amelyek megoldásainak egyes részletei többször ismétlődnek, vagy a program jól elkülöníthető részekre bontható, és ezeket szubrutinokkal is megoldani!

10.4 feladat: Készítsünk olyan programot, amelyik kiír öt darab, keretben elhelyezett lottószámot!

10.5 feladat: Készítsünk olyan szubrutint, amelyik valamilyen alakzatot rajzol a képernyőre! Ezt a szubrutint felhasználva írjunk olyan programot, amelyben tetszőleges billentyűt leütve, a képernyőn véletlen helyen, véletlen nagyságban vagy kirajzolja az alakzatot, vagy az alakzatnak megfelelő helyet töröl!

Kapcsoló utasítások

Futtassuk le a következő példaprogramot!

10.6 példaprogram

```
10 CLS
20 INPUT I
30 ON I GOTO 100,200,300 : REM KAPCSOLO
40 STOP
100 PRINT"I=1" : STOP
200 PRINT"I=2" : STOP
300 PRINT"I=3" : STOP
```

Tapasztalhattuk, hogyha az I változó értéke 1, 2 vagy 3, akkor a vezérlés rendre a 100-as, 200-as, vagy a 300-as sorra adódik át. Ha az I változó értéke nem egész szám, akkor annak egészrészét veszi, ha negatív, akkor a program hibajelzéssel leáll, ha pedig nulla vagy nagyobb mint három, akkor a 40-es sorral folytatódik a program.

Ha a 10.6 példaprogram 100-as, 200-as, 300-as soraiban szubrutinok kezdődnek, akkor azokat a

30 ON I GOSUB 100, 200, 300

utasítással hívhatjuk meg. Ebben az esetben a szubrutint befejező RETURN utasítás után mindig a 40-es sorra adódik vissza a vezérlés.

10.6 feladat: Írjunk olyan programot, amelyik beolvas egy szót, majd
az [1] billentyű lenyomására kiírja annak hosszát,
a [2] billentyű lenyomására kiírja a szó magánhangzóit,
a [3] billentyű lenyomására kiírja a magánhangzók számát,
a [4] billentyű lenyomására kiírja a szót visszafelé,
az [5] billentyűt lenyomva pedig új szó megadását várja! A lehetséges választásokról állandóan tájékoztató legyen a képernyőn.

10.7 feladat: Hol a hiba?

```
10 CLS
20 FOR I=0 TO 100
30   PRINT@0,I;
40   GOSUB 60 : GOSUB 80
50 NEXT
60 SET(11,21)
70 RETURN
80 RESET(11,21)
90 RETURN
100 END
```

10.8 feladat: Hol a hiba?

```
10 CLS
20 GOSUB 30 : GOTO 20
30 SET(22,31)
40 GOSUB 60
50 RETURN
60 RESET(22,31)
70 RETURN
```

Gyakorlatok a tizedik fejezet anyagához

10.1 gyakorlat: Készítsünk olyan programot, amelyik beolvas *N* darab számot, csökkenő sorrendbe rendezi és kiírja a képernyőre őket! Az *N* bemenő adat legyen.

10.2 gyakorlat: Készítsünk programot, amelynek segítségével éves költségvetésünket tartjuk nyilván!

10.3 gyakorlat: Mozogjon három négyzet egymástól függetlenül véletlenszerűen a képernyőn!

10.4 gyakorlat: Jelenjen meg a képernyőn öt ajtó. Ezeket véletlenszerűen nyitják és csukják. Készítsük el ezt a programot!

10.5 gyakorlat: Készítsünk órát a képernyőre szubrutinok felhasználásával!

10.6 gyakorlat: Mozogjon egy bábu a képernyőn!

10.7 gyakorlat: Véletlenszámokkal töltsünk fel egy 6 × 6-os tömböt. Válasszuk ki minden sorból a legkisebb elemet, majd közülük írjuk ki a legnagyobbat, ezután minden oszlopból válasszuk ki a legnagyobb elemet, majd ezek közül írjuk ki a legkisebbet!

10.8 gyakorlat: Készítsünk olyan programot, amelyik bekéri egy háromszög három csúcspontjának a koordinátáit, és kirajzolja a háromszöget!

10.9 gyakorlat: Írjunk olyan programot, amelynek segítségével két 50 jegyű számot össze tudunk adni!

10.10 gyakorlat: Írjunk olyan programot, amelynek segítségével két 20 jegyű számot össze tudunk szorozni!

10.11 gyakorlat: Készítsünk olyan programot, amelyiknek ha megadunk egy születési dátumot és a mai dátumot, akkor kiszámolja a két időpont között eltelt napok számát!

A tizedik fejezet feladatainak egy lehetséges megoldása

10.1 feladat

```
10 REM SAKKTABLAT  RAJZOLO PROGRAM
20 CLS
30 DEFINT A-Z
40 REM A RAJZOLAST VEZERLO RESZ
50 FOR A=0 TO 70 STEP 10
60   FOR B=0 TO 35 STEP 5
70     X=A/20 : Y=B/2
80     IF A=20*X AND B=2*Y THEN 100
90     IF A>20*X AND B>2*Y THEN 100 ELSE 150
100     REM NEGYZETRAJZOLO
110     FOR I=0 TO 9
120       FOR K=0 TO 4
130         SET(A+I,B+K)
140     NEXT K,I
150 NEXT B,A
160 REM BABUMOZGATO RESZ
170 A=(RND(8)-1)*10 : B=(RND(8)-1)*5
180 REM BABURAJZOLO
190 RESTORE
200 FOR I=1 TO 9
210   READ X,Y
220   IF POINT(A+X,B+Y) THEN RESET(A+X,B+Y) ELSE SET(A+X,B+Y)
230 NEXT I
240 FOR I=1 TO 200 : NEXT
250 REM BABURAJZOLO
260 RESTORE
270 FOR I=1 TO 9
280   READ X,Y
290   IF POINT(A+X,B+Y) THEN RESET(A+X,B+Y) ELSE SET(A+X,B+Y)
300 NEXT I
310 GOTO 170
320 DATA 4,1,5,1,6,1,4,2,5,2,3,3,4,3,5,3,6,3
```

10.2 feladat

```
10 REM FOPROGRAM
20 CLS
30 IF RND(2)=1 THEN GOSUB 80 ELSE GOSUB 120
40 PRINT@970,"FOLYTATOD ? (I/N) ";
50 A$=INKEY$
60 IF A$="I" THEN 20
70 IF A$="N" THEN STOP ELSE 50
80 REM VONALRAJZOLO SZUBRUTIN
90 X=RND(45)-1
100 FOR Y=0 TO 127 : SET(Y,X) : NEXT Y
110 RETURN
120 REM TEGLALAPRAJZOLO SZUBRUTIN
130 X=RND(45)-1 : Y=RND(128)-1
140 FOR I=0 TO X : FOR J=0 TO Y : SET(J,I) : NEXT J,I
150 RETURN
```

10.3 feladat

```
10 CLS
20 DEFINT I,J,K
30 PRINT@85,"MELYIK BETUT IRJAM KI :  T-H-TH-HT  "; : INPUT A$
40 IF A$="T" THEN J=23 : GOSUB 90 : GOTO 80
50 IF A$="H" THEN J=23 : GOSUB 130 : GOTO 80
60 IF A$="TH" THEN J=8 : GOSUB 90 : J=J+30 : GOSUB 130 : GOTO 80
70 IF A$="HT" THEN J=8 : GOSUB 130 : J=J+30 : GOSUB 90 : GOTO 80
              ELSE 20
80 GOTO 80
90 REM A T BETU
100 FOR K=1 TO 3 : PRINT@(3+K)*64+J,STRING$(22,191); : NEXT K
110 FOR K=4 TO 9 : PRINT@(3+K)*64+J+5,STRING$(12,191); : NEXT K
120 RETURN
130 REM A H BETU
140 FOR K=1 TO 9
150   PRINT@(3+K)*64+J,STRING$(5,191); :
      PRINT@(3+K)*64+J+17,STRING$(5,191);
160 NEXT K
180 FOR K=4 TO 6 : PRINT@(3+K)*64+J+5,STRING$(12,191); : NEXT K
200 RETURN
```

10.4 feladat

```
10 CLS
20 DEFINT A,I,J,K
30 PRINT@222,"A LOTTOSZAMOK :";
40 FOR I=1 TO 5 : GOSUB 60 : GOSUB 100 : PRINT@651+I*8,A(I); : NEXT
50 GOTO 50
60 REM KERETRAJZOLAS
70 FOR J=1 TO 10 : SET(20+I*16+J,25) : SET(20+I*16+J,35) : NEXT
80 FOR J=1 TO 9 : SET(21+I*16,25+J) : SET(30+I*16,25+J) : NEXT
90 RETURN
100 REM AZ I. LOTTOSZAM GENERALASA
110 A(I)=RND(90)
120 FOR  K=0 TO I-1
130   IF A(K)=A(I) THEN 110
140 NEXT
150 RETURN
```

10.5 feladat

```
10 CLS
20 DEFINT I,J,K,X
30 IF INKEY$="" THEN 30
40 I=RND(830) : J=RND(2) : K=RND(6)
50 IF J=1 THEN J=191 ELSE J=32
60 IF I+K*64+K>1022 THEN 40
70 GOSUB 90
80 GOTO 30
90 REM RAJZ
100 FOR X=1 TO K
110   PRINT@I+(X-1)*64,STRING$(K,J);
120 NEXT
130 RETURN
```

10.6 feladat

```
10 DEFINT I-K
20 Z$="AEIOU"
30 CLS
40 INPUT"MELYIK SZOT VIZSGALJAM ";S$
50 PRINT@0,CHR$(255) : PRINT@5,S$;
60 PRINT@13*64,"A SZO HOSSZA ----------- 1,";
   " A SZO MAGANHANGZOI ------------ 2,"; :
   PRINT@14*64,"A MAGANHANGZOK SZAMA --- 3,";
   " A SZO VISSZAFELE -------------- 4,";  :
   PRINT@15*64,"UJ SZO ---------------- 5";
70 GOSUB 240
80 IF I=5 THEN 30
90 ON I GOSUB 290,110,140,320
100 GOTO 70
110 REM A SZO MAGANHANGZOI
120 GOSUB 170 : PRINT@5*64+5,CHR$(255) : PRINT@5*64+5,B$;
130 RETURN
140 REM A MAGANHANGZOK SZAMA
150 GOSUB 170 : PRINT@5*64+5,CHR$(255) : PRINT@5*64+5,LEN(B$);
160 RETURN
170 REM
180 J1=LEN(S$) : J2=LEN(Z$) : B$=""
190 FOR K1=1 TO J1
200   FOR K2=1 TO J2
210     IF MID$(S$,K1,1)=MID$(Z$,K2,1) THEN B$=B$+MID$(S$,K1,1)
220 NEXT K2,K1
230 RETURN
240 REM INFORMACIO A BILLENTYUZETROL
250 A$=INKEY$ : IF A$="" THEN 250
260 I=VAL(A$)
270 IF I<1 OR I>5 THEN 250
280 RETURN
290 REM EGY SZO BETUINEK SZAMA
300 PRINT@5*64+5,CHR$(255) : PRINT@5*64+5,LEN(S$);
310 RETURN
320 REM A SZO MEGFORDITASA
330 K=LEN(S$) : A$=""
340 FOR J=1 TO K
350   A$=LEFT$(S$,1)+A$ : S$=S$+LEFT$(S$,1) : S$=RIGHT$(S$,K)
360 NEXT
370 PRINT@5*64+5,CHR$(255) : PRINT@5*64+5,A$;
380 RETURN
```

10.7 feladat

Az END (vagy esetleg egy STOP) utasítást az 55-ös sorba kell írni.

10.8 feladat

S E H O L ! ! !

Tizenegyedik fejezet

A SZÁMÍTÓGÉP LELKIVILÁGA

PEEK, POKE

A számítógép memóriája „rekeszekre" van osztva. Minden rekesznek van címe, és minden rekeszben található egy 256-nál kisebb nemnegatív egész szám. A **PEEK(N)** függvényutasítás segítségével ki lehet olvasni az N című memóriarekesz tartalmát.

A **POKE** utasítás alkalmazásával pedig a memória bizonyos rekeszeibe írhatunk értékeket. A

POKE N,A [NL]

parancs hatására az N című rekesz tartalma az A szám lesz. Érdemes megjegyezni, hogy a 1536Ø, 15361, ..., 16383 című rekeszek tartalma a képernyő Ø. – 1Ø23. karakterpozíciójával van szoros kapcsolatban. Ha például az előbbi parancsban az N értékére

$$15360 \leqq N \leqq 16383$$

és az A értékére

$$32 \leqq A \leqq 192$$

teljesül, akkor a képernyő N – 1536Ø-adik karakterpozícióján megjelenik az A ASCII-kódú karakter.

11.1 feladat: Készítsünk olyan programot, amelyik a képernyő kívánt helyén egy kívánt karaktert kigyújt!

A POKE utasítás egyik előnye a PRINT utasítással szemben az, hogyha segítségével a képernyőn az 1Ø23. karakterpozícióba írunk, akkor sem emel sort.

11.2 feladat: Rajzoltassunk szimmetrikus, lehető legnagyobb keretet a képernyőre!

Hasznos tudni, hogy a 16537-es című memóriarekesz az INKEY függvény tárolóterülete.

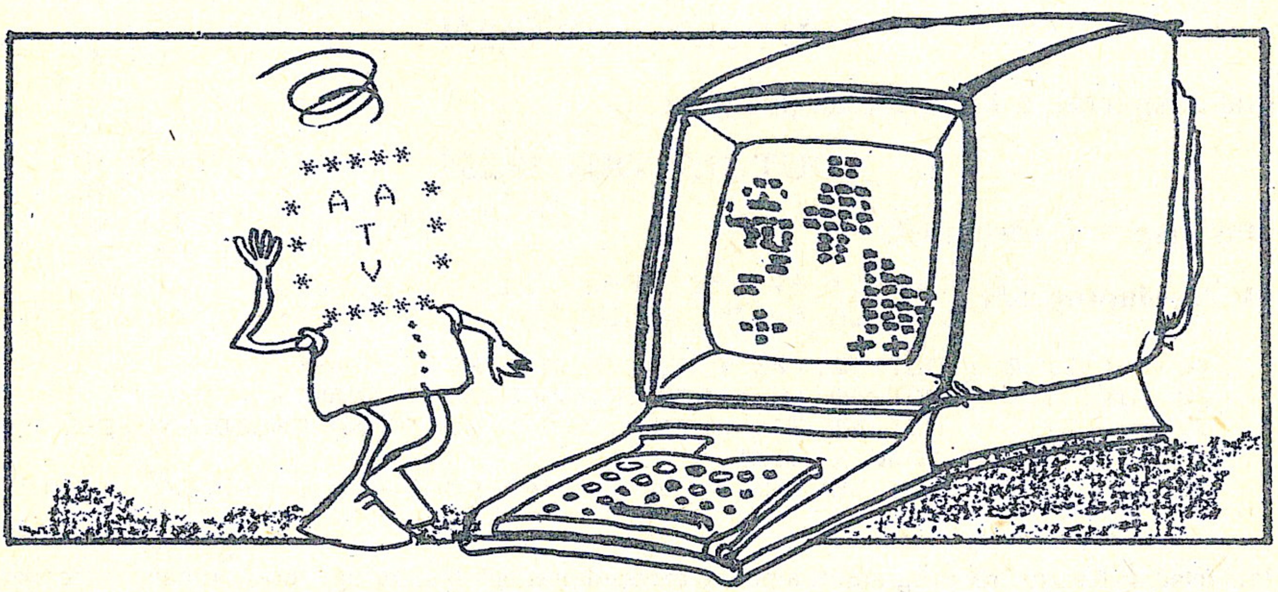
11.1 példaprogram

```
10 CLS
20 FOR I=15360 TO 15519
30   POKE I,I-15328
40 NEXT I
50 IF PEEK(16537)=0 THEN 50 ELSE POKE 16537,0
60 GOTO 10
```

Egy kis ízelítő a számítógép grafikájából.

11.2 példaprogram

```
10 X=RND(63)+131
20 IF RND(2)-1 THEN J=15360 : L=16383 : S=RND(100)
     ELSE J=16383 : L=15360 : S=-RND(100)
30 CLS
40 FOR M=J TO L STEP S
50   POKE M,X
60 NEXT
70 GOTO 10
```



Hanggenerátor

Az iskolaszámítógép a beépített hanggenerátor segítségével hangokat is tud adni. Ezekből összeállíthatunk egyszerű dallamot, figyelemfelkeltő és kísérő effektusokat. Így programjainkat színesebbé, tartalmasabbá tehetjük.

A hanggenerátor vezérlése az OUT utasítással történhet.

Egy hang megszólaltatásához három teendőnk van, három ún. *regiszter*t kell beállítanunk. Ezek az RØ, R7, R8 regiszterek. (A számítógépen 13 regiszter áll

kapcsolatban a hanggenerátorral, de számunkra most csak az említett három lesz fontos.) Az

OUT 31,X : OUT 3Ø,Y

utasításpár hatása a következő:

Az OUT 31,X utasítás kiválasztja az RX regisztert. Az OUT 3Ø,Y utasítás pedig az RX regiszterbe beírva az Y értékének egészrészét, beállítja azt.

A három említett regiszter működése a következő:

Az **RØ** regiszter a *hangmagasság* beállítására szolgál. Az Y értékének 256-nál kisebb nemnegatív számnak kell lenni. Kisebb Y értékhez magasabb hang tartozik.

Az **R8** regiszterrel a megszólaló hang *erősségé*t lehet szabályozni. 16 erősségi fokozatot állíthatunk be úgy, hogy a regiszterbe a Ø, 1, 2, ..., 15 számok valamelyikét írjuk.

Az **R7** regiszter segítségével tudjuk a számítógép ún. „A" hangcsatornáját *be-* és *kikapcsol*ni. (A további két – „B" és „C" – csatornával nem foglalkozunk.) A bekapcsolás az

OUT 31,7 : OUT 3Ø,254

utasításpárral, a kikapcsolás pedig az

OUT 31,7 : OUT 3Ø,255

utasításpárral történhet.

11.3 példaprogram

```
10 OUT 31,0 : OUT 30,100 : REM A HANG BEALLITASA
20 OUT 31,8 : OUT 30,15 : REM A HANGERO MAXIMALIS
30 OUT 31,7 : OUT 30,254 : REM AZ "A" CSATORNA ENGEDELYEZESE
40 FOR I=0 TO 1000 : NEXT
50 OUT 31,7 : OUT 30,255 : REM AZ "A" CSATORNA LEZARASA (CSEND)
```

11.3 feladat: Készítsünk programot, amelyik megszólaltat egy hangot úgy, hogy annak hangerejét a maximumtól a minimumig csökkenti!

11.4 feladat: Írjunk olyan programot, amelyik a hangok magasságának növekvő sorrendjében szólaltat meg hangokat!

Érdemes megjegyezni, hogy a természetes hangskála alaphangjainak a következő Y értékek felelnek meg:

c–2Ø8, *d*–185, *e*–165, *f*–155, *g*–139, *a*–124, *h*–11Ø, *felső c*–1Ø4

11.5 feladat: Szólaltassa meg a számítógép a természetes hangskála alaphangjait egymás után, és a megszólaló hanggal egy időben jelenjen meg a képernyő közepén annak neve is!

A kétoktávnyi teljes hangskála megfelelő értékei:

c–2Ø8, *cisz*–195, *d*–185, *disz*–175, *e*–165, *f*–155, *fisz*–146, *g*–139, *gisz*–13Ø, *a*–124, *aisz*–117, *h*–11Ø;
1Ø4, 98, 92, 87, 83, 78, 73, 7Ø, 65, 62, 59, 55

11.6 feladat: A számítógép véletlenszerűen szólaltassa meg az alapskála egy hangját, majd ezt követően az egy oktávval magasabbat! A megszólaló hanggal egy időben jelenjen meg a képernyő közepén a neve is! Az [I] billentyű lenyomására ezt ismételje, más billentyű lenyomásakor pedig álljon le a program.

Végezetül a számítógép zenéje:

11.4 példaprogram

```
10 DEFINT H-N : DIM H(23)
20 OUT 31,8 : OUT 30,15
30 OUT 31,7 : OUT 30,254
40 FOR I=0 TO 23
50   READ H(I)
60 NEXT I
70 K=RND(24)-1
80 CLS
90 OUT 31,0 : OUT 30,H(K)
100 GOTO 70
110 DATA 208,195,185,175,165,155,146,139,130,124,117,110
120 DATA 104,98,92,87,83,78,73,70,65,62,59,55
```

11.7 feladat: Hol a hiba?

```
10 CLS
20 FOR I=15360 TO 16397 STEP 7
30   POKE I,0
40 NEXT
50 GOTO 10
```

11.8 feladat: Hol a hiba?

```
10 CLS
20 OUT 31,7 : OUT 30,254
30 OUT 31,8 : OUT 30,10
40 OUT 31,0 : OUT 30,110
50 PRINT@6*64+5,"A PROGRAM VEGE"
60 END
```

Gyakorlatok a tizenegyedik fejezet anyagához

11.1 gyakorlat: Írjunk programot, amelynek segítségével a kurzor a képernyőn bárhova mozgatható, de a mozgatással a képernyőn semmi sem törlődik!

11.2 gyakorlat: Módosítsuk a 11.1 gyakorlatra adott megoldásunkat úgy, mintha a képernyő szemközti oldalai össze lennének ragasztva!

11.3 gyakorlat: Írjunk olyan programot, amelynek segítségével a billentyűzetről bármilyen ábrát rajzoltathatunk a képernyőre! (Használjuk a teljes, 32–191 ASCII-kódú karakterkészletet!)

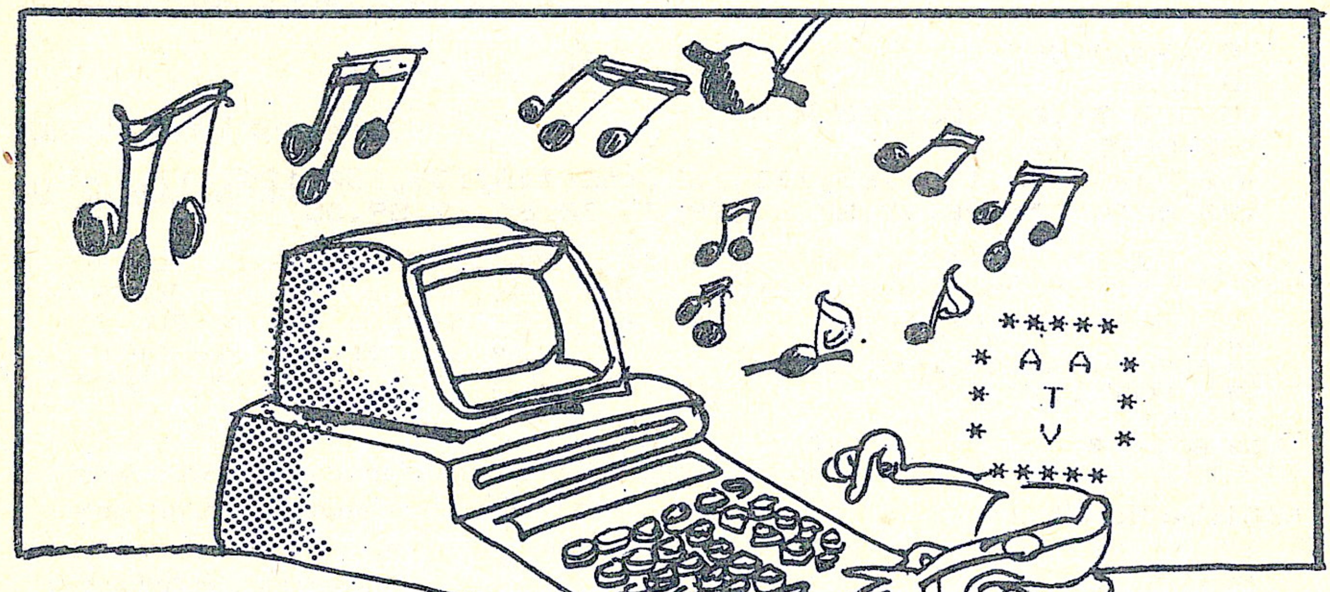
11.4 gyakorlat: A számítógép képernyőjén megjelenik egy véletlen karakterlánc. Célunk az, hogy ebből a billentyűzetről vezérelve értelmes szót alakítsunk ki. Minden betűcsere egy lépésnek számít. Miután megkaptuk az értelmes szót, a gép írja ki a betűcserék számát.

11.5 gyakorlat: Készítsünk olyan programot, amelynek segítségével egy adott hangot egy adott ideig szólaltathatunk meg!

11.6 gyakorlat: Írjunk olyan programot, amelyik egy ismert népdal dallamát lejátssza!

11.7 gyakorlat: A számítógép képernyőjén jelenjenek meg keretbe foglalva a természetes hangskála alaphangjainak nevei. Ha lenyomunk egy hangnak megfelelő billentyűt, akkor az a hang szólaljon meg, és villogjon a neve!
Készítsük el ezt a programot!

11.8 gyakorlat: Egészítsük ki a 11.2 példaprogramot úgy, hogy ne csak rajzoljon, hanem zenéljen is a számítógép!



A tizenegyedik fejezet feladatainak egy lehetséges megoldása

11.1 feladat

```
1Ø DEFINT X,Y
2Ø CLS
3Ø INPUT"A KARAKTERPOZICIO ";X : INPUT"A KARAKTER KODJA ";Y
4Ø IF X<Ø OR X>1Ø23 THEN 2Ø
5Ø CLS
6Ø POKE X+1536Ø,Y
7Ø IF INKEY$="" THEN 7Ø ELSE 2Ø
```

11.2 feladat

```
1Ø CLS
2Ø X=1536Ø
3Ø FOR J=Ø TO 15
4Ø   POKE X+J*4,14Ø : POKE X+J*64-1,14Ø : POKE X+J*64,14Ø :
     POKE 1632Ø+J*4,14Ø
5Ø NEXT
6Ø POKE 16383,14Ø
7Ø GOTO 7Ø
```

11.3 feladat

```
1Ø OUT 31,Ø : OUT 3Ø,1ØØ
2Ø OUT 31,7 : OUT 3Ø,254
3Ø FOR J=15 TO Ø STEP-1
4Ø   OUT 31,8 : OUT 3Ø,J : REM A HANGERO FOKOZATOSAN CSOKKEN
5Ø   FOR I=Ø TO 1ØØ : NEXT I
6Ø NEXT J
7Ø OUT 31,7 : OUT 3Ø,255
```

11.4 feladat

```
1Ø OUT 31,8 : OUT 3Ø,1Ø
2Ø OUT 31,7 : OUT 3Ø,254
3Ø FOR J=255 TO Ø STEP -1
4Ø   OUT 31,Ø : OUT 3Ø,J : REM A HANGMAGASSAG NO
5Ø NEXT J
6Ø OUT 31,7 : OUT 3Ø,255
```

11.5 feladat

```
1Ø CLS
2Ø OUT 31,Ø : OUT 3Ø,Ø
3Ø OUT 31,8 : OUT 3Ø,15
4Ø OUT 31,7 : OUT 3Ø,254
5Ø FOR J=1 TO 8
6Ø   READ H$,H
7Ø   PRINT@8*64+3Ø,H$
8Ø   OUT 31,Ø : OUT 3Ø,H : REM A SKALA MEGFELELO HANGJA
9Ø   FOR I=Ø TO 333 : NEXT I
1ØØ NEXT J
11Ø OUT 31,7 : OUT 3Ø,255
12Ø STOP
13Ø DATA C,2Ø8,D,185,E,165,F,155,G,139,A,124,H,11Ø,c,1Ø4
```

A 2Ø-as sor az RØ regiszter nullázását jelenti, ami azért célszerű, mert így a program futtatásakor először a skála kívánt hangját szólaltatja meg.

11.6 feladat

```
10 DEFINT H-N : DIM H(23),A$(11)
20 OUT 31,0 : OUT 30,0
30 OUT 31,8 : OUT 30,15
40 OUT 31,7 : OUT 30,254
50 OUT 31,0
60 FOR I=0 TO 11 : READ A$(I),H(I) : NEXT I
90 FOR I=12 TO 23 : READ H(I) : NEXT I
120 FL=0 : K=RND(12)-1 : HA=H(K)
130 CLS
140 PRINT@8*64+32,A$(K)
150 OUT 30,HA
160 FOR I=0 TO 1000 : NEXT I
170 IF FL=1 THEN OUT 30,0 : GOTO 190
180 FL=1 : HA=H(K+12) : PRINT@8*64+24,"FELSO "; : GOTO 150
190 PRINT@15*64,"FOLYTASSAM ?  (I/N)";
200 T$=INKEY$
210 IF T$="" THEN 200
220 IF T$="I" THEN 120
230 STOP
240 DATA C,208,CISZ,195,D,185,DISZ,175,E,165,F,155,FISZ,146
250 DATA G,139,GISZ,130,A,124,AISZ,117,H,110
260 DATA 104,98,92,87,83,78,73,70,65,62,59,55
```

Az 5Ø-es valamint a 15Ø-es és a 17Ø-es sorokban látható, hogy az

OUT 31,X   és az   OUT 3Ø,Y

utasításokat nem kötelező közvetlenül egymás után írni. A 17Ø-es sorban levő OUT3Ø,Ø utasítás az A hangcsatornán *csendet* állít be.

11.7 feladat

Ha a memóriának olyan rekeszébe írunk, amelyiknek nem ismerjük a jelentését, akkor sok meglepetés érhet bennünket!!!

11.8 feladat

A hanggenerátort vagy a RESET gombbal, vagy az OUT utasítással, vagy pedig a számítógép kikapcsolásával lehet elhallgattatni.

Tizenkettedik fejezet

A GYAKORLATOK EGY LEHETSÉGES MEGOLDÁSA

Első fejezet

```
10 REM ..... 1.1 G Y A K O R L A T
20 CLS
30 SET(40,0) : SET(40,30) : SET(100,15)
40 SET(20,15) : SET(80,0) : SET(80,30)
50 RESET(20,15) : RESET(80,0) : RESET(80,30)
60 GOTO 40
```

```
10 REM ..... 1.2 G Y A K O R L A T
20 CLS
30 SET(10,3) : SET(9,4) : SET(8,5) : SET(7,6) : SET(6,7) :
   SET(8,7) : SET(10,7) : SET(12,7) : SET(14,7) : SET(11,4) :
   SET(12,5) : SET(13,6)
40 GOTO 40
```

```
10 REM ..... 1.3 G Y A K O R L A T
20 CLS
30 INPUT" A PONT KOORDINATAI ";X,Y
40 CLS
50 SET(X,Y) : SET(X-1,Y+1) : SET(X-2,Y+2) : SET(X-3,Y+3) :
   SET(X-4,Y+4) : SET(X-2,Y+4) : SET(X,Y+4) : SET(X+2,Y+4) :
   SET(X+4,Y+4) : SET(X+1,Y+1) : SET(X+2,Y+2) : SET(X+3,Y+3)
60 GOTO 60
```

```
10 REM ..... 1.4 G Y A K O R L A T
20 CLS
30 X=0
40 SET(X,11)
50 X=X+1
60 GOTO 40
```

```
10 REM ..... 1.5 G Y A K O R L A T
20 CLS
30 SET(1,1) : SET(10,1) : SET(100,1) : SET(100,10) :
   SET(100,30) : SET(80,30) : SET(60,30) : SET(40,30) :
   SET(30,30) : SET(20,30) : SET(10,30) : SET(0,30) :
   SET(0,40) : SET(10,40) : SET(15,40)
40 RESET(15,40) : RESET(10,40) : RESET(0,40) : RESET(0,30) :
   RESET(10,30) : RESET(20,30) : RESET(30,30) : RESET(40,30) :
   RESET(60,30) : RESET(80,30) : RESET(100,30) :
   RESET(100,10) : RESET(100,1) : RESET(10,1) : RESET(1,1)
50 GOTO 30
```

```
10 REM ..... 1.6 G Y A K O R L A T
20 CLS
30 SET(5,6) : SET(5,7) : SET(5,8) : SET(5,9) : SET(6,9) :
   SET(7,9) : SET(8,9) : SET(9,9)
40 GOTO 40
```

```
10 REM ..... 1.7 G Y A K O R L A T
20 CLS
30 INPUT"AZ ELSO PONT KOORDINATAI ";X1,Y1 :
   INPUT"A MASODIK PONT KOORDINATAI ";X2,Y2
40 CLS
50 SET(X1,Y1) : SET(X1,Y1) : SET(X1,Y1) : SET(X1,Y1) :
   PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:
   PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
60 CLS
70 SET(X2,Y2) : SET(X2,Y2) : SET(X2,Y2) : SET(X2,Y2) :
   PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:
   PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
80 GOTO 40
```

```
10 REM ..... 1.8 G Y A K O R L A T
20 CLS
30 X=0 : Y=0
40 X=X+3 : Y=Y+1
50 SET(X,0) : SET(X,10) : RESET(X,0) : RESET(X,10)
60 SET(0,Y) : SET(10,Y) : RESET(0,Y) : RESET(10,Y)
70 GOTO 40
```

```
10 REM ..... 1.9 G Y A K O R L A T
20 CLS
30 X=1 : Y=1 : A=124
40 SET(X,Y) : RESET(X,Y) : SET(A,Y) : RESET(A-3,Y-1)
50 X=X+3 : Y=Y+1 : A=A-3
60 GOTO 40
```

```
10 REM ..... 1.10 G Y A K O R L A T
20 CLS
30 INPUT"A KET KOORDINATA ";X1,Y1
40 CLS
50 X=X1 : A=X : Y=Y1 : B=Y
60 X=X+3 : Y=Y+1 : A=A-3 : B=B-1
70 SET(X,Y1) : SET(X1,Y) : SET(A,Y1) : SET(X1,B)
80 PRINT : PRINT : PRINT : PRINT : PRINT
90 CLS
100 GOTO 60
```

```
10 REM ..... 1.11 G Y A K O R L A T
20 CLS
30 X=0 : Y=0
40 SET(X,Y) : SET(X+10,Y) : SET(X+10,Y+5) : SET(X,Y+5)
50 X=X+3 : Y=Y+1
    S
    TO 40
```

```
10 REM ..... 1.12 G Y A K O R L A T
20 CLS
30 X=0 : A=127
40 SET(X,10) : SET(A,10)
50 X=X+1 : A=A-1
60 RESET(X-1,10) : RESET(A+1,10)
70 GOTO 40
```

```
10 REM ..... 1.13 G Y A K O R L A T
20 CLS
30 X=63 : Y=23 : A=63 : B=23
40 RESET(X-3,Y-1) : RESET(A+3,B+1) : RESET(63,Y-1) :
   RESET(X-3,23) : RESET(63,B+1) : RESET(A+3,23) :
   RESET(X-3,B+1) : RESET(A+3,Y-1)
50 SET(X,Y) : SET(A,B) : SET(63,Y) : SET(X,23) : SET(63,B) :
   SET(A,23) : SET(X,B) : SET(A,Y)
60 X=X+3 : Y=Y+1 : A=A-3 : B=B-1
70 GOTO 40

10 REM ..... 1.14 G Y A K O R L A T
20 CLS
30 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:
   PRINT"                              SZIA !"
40 GOTO 40

10 REM ..... 1.15 G Y A K O R L A T
20 CLS
30 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:
   PRINT"                        V I L L O G"
40 X=X-1 : X=X+1 : X=X+1 : X=X-1 : X=X-1 : X=X+1 : X=0 : X=0
50 X=X+1 : X=X-1 : X=X+1 : X=X-1 : X=X+1 : X=X-1
60 GOTO 20

10 REM ..... 1.16 G Y A K O R L A T
20 PRINT "*";
30 GOTO 20

10 REM ..... 1.17 G Y A K O R L A T
20 CLS
30 PRINT"    A" : PRINT"   A A" : PRINT"  A   A" :
   PRINT" A A A A" : PRINT"A       A"
40 GOTO 40

10 REM ..... 1.18 G Y A K O R L A T
20 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:
   PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
30 PRINT"==============================================================;
40 GOTO 20

10 REM ..... 1.19 G Y A K O R L A T
20 PRINT"                                              /"
30 PRINT"                                            /"
40 PRINT"                                          /"
50 PRINT"                                        /"
60 PRINT"                                      /"
70 PRINT"                                    /"
80 PRINT"                                  /"
90 PRINT"                                /"
100 PRINT"                              /"
110 PRINT"                            /"
120 PRINT"                          /"
130 PRINT"                        /"
140 PRINT"                      /"
150 PRINT"                    /"
160 PRINT"                  /"
170 PRINT"                /"
180 PRINT"              /"
190 PRINT"            /"
200 PRINT"          /"
210 PRINT"        /"
220 PRINT"      /"
230 PRINT"    /"
240 PRINT"  /"
250 GOTO 20
```

```
10 REM ..... 1.20 G Y A K O R L A T
20 INPUT A,B,C
30 PRINT : PRINT : PRINT
40 PRINT"A=";A,"B=";B,"C=";C
50 PRINT : PRINT
60 PRINT"A=";A : PRINT"B=";B : PRINT"C=";C
70 PRINT : PRINT : PRINT : PRINT
80 GOTO 20
```

Második fejezet

```
10 REM ..... 2.1 G Y A K O R L A T
20 CLS
30 FOR X=5 TO 25 STEP 5
40    FOR Y=5 TO 25 : SET(X,Y) : NEXT Y
50 NEXT X
60 CLS
70 FOR X=5 TO 25
80    FOR Y=5 TO 25 STEP 5 : SET(X,Y) : NEXT Y
90 NEXT X
```

```
10 REM ..... 2.2 G Y A K O R L A T
20 CLS
30 FOR X=10 TO 100 STEP 10
40    FOR Y=0 TO 47 : SET(X,Y) : NEXT Y
50 NEXT X
60 FOR X=0 TO 127
70    FOR Y=4 TO 30 STEP 8 : SET(X,Y) : NEXT Y
80 NEXT X
90 FOR X=10 TO 90 : SET(X,45) : NEXT X
100 GOTO 100
```

```
10 REM ..... 2.3 G Y A K O R L A T
20 CLS
30 FOR K=1 TO 10
40    PRINTK;"---------------------------------"
50 NEXT K
60 GOTO 60
```

```
10 REM ..... 2.4 G Y A K O R L A T
20 CLS
30 FOR K=0 TO 20
40    SET(63,23+K) : SET(63,23-K) :
      SET(63+K+K+K,23) : SET(63-K-K-K,23) :
      SET(63+K+K+K,23+K) : SET(63+K+K+K,23-K) :
      SET(63-K-K-K,23+K) : SET(63-K-K-K,23-K)
50 NEXT K
60 GOTO 60
```

```
10 REM ..... 2.5 G Y A K O R L A T
20 CLS
30 FOR Y=3 TO 35 STEP 6
40    PRINT"-----------------------------------------"
50    PRINT
60    FOR X=0 TO 59 : SET(X,Y) : NEXT X
70 NEXT Y
80 PRINT
90 STOP
```

```
10 REM ..... 2.6 G Y A K O R L A T
20 CLS
30 FOR X=0 TO 127 : SET(X,0) : SET(X,47) : NEXT X
40 FOR Y=0 TO 47 : SET(0,Y) : SET(127,Y) : NEXT Y
50 GOTO 50

10 REM ..... 2.7 G Y A K O R L A T
20 PRINT"* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ";
30 FOR K=1 TO 13
40 PRINT"*                                                              * ";
50 NEXT
60 PRINT"* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ";
70 GOTO 70

10 REM ..... 2.8 G Y A K O R L A T
20 CLS
30 PRINT"ELSO SOR" : PRINT : PRINT"HARMADIK SOR" : PRINT :
   PRINT"OTODIK SOR"
40 FOR Y=3 TO 20 STEP 6
50   FOR X=0 TO 36 : SET(X,Y) : NEXT X
60 NEXT Y
70 PRINT : PRINT

10 REM ..... 2.9 G Y A K O R L A T
20 CLS
30 FOR J=140 TO 10 STEP -10 : PRINT J : NEXT J
40 PRINT
50 PRINT"  0   10   20   30   40   50   60   70   80   90   100   110";
60 FOR X=0 TO 100 : SET(X,43) : NEXT X
70 FOR Y=0 TO 47 : SET(9,Y) : NEXT Y
80 GOTO 80

10 REM ..... 2.10 G Y A K O R L A T
20 CLS
30 PRINT"CSAK POZITIV KOORDINATAJU PONTOKAT
     TUDOK ABRAZOLNI !"
40 INPUT X,Y
50 IF X>0 THEN 60 ELSE 20
60 IF Y>0 THEN 70 ELSE 20
70 FOR K=1 TO 15
80   IF K=7 THEN PRINT Y ELSE PRINT
90 NEXT K
100 PRINT" 0";"                                   ";X;
110 FOR X=0 TO 127 : SET(X,42) : NEXT X
120 FOR Y=0 TO 47 : SET(10,Y) : NEXT Y
130 SET(60,18)
140 GOTO 140

10 REM ..... 2.11 G Y A K O R L A T
20 CLS
30 PRINT : PRINT : PRINT : PRINT : PRINT : PRINT : PRINT
40 PRINT"                              MASSALHANGZOK"
50 FOR I=1 TO 300 : NEXT
60 SET(59,21) : SET(65,21) : SET(71,21) : SET(79,21)
70 GOTO 70

10 REM ..... 2.12 G Y A K O R L A T
20 CLS
30 PRINT"* * * * * * * * * * * * *" :
   PRINT"*       *       *       *" :
   PRINT"*   1   *   2   *   3   *" :
   PRINT"*       *       *       *" :
   PRINT"* * * * * * * * * * * * *"
40 PRINT : PRINT
```

```
10 REM ..... 2.13 G Y A K O R L A T
20 X=0
30 CLS
40 FOR J=0 TO X : PRINT : NEXT J
50 PRINT"+++++"
60 FOR K=1 TO 1000 : NEXT
70 X=X+1
80 GOTO 30

10 REM ..... 2.14 G Y A K O R L A T
20 X=1
30 CLS
40 PRINT X
50 FOR I=1 TO 1000 : NEXT
60 X=X+1
70 GOTO 30

10 REM ..... 2.15 G Y A K O R L A T
20 CLS
30 FOR Y=1 TO 5 : SET(10,Y) : NEXT Y
40 FOR K=1 TO 300 : NEXT
50 CLS
60 FOR Y=1 TO 5 : SET(120,Y) : NEXT Y
70 FOR K=1 TO 300 : NEXT
80 GOTO 20

10 REM ..... 2.16 G Y A K O R L A T
20 X=0 : Y=0
30 CLS
40 PRINT X ; " ORA",Y;" PERC"
50 FOR K=1 TO 200 : NEXT
60 Y=Y+1
70 IF Y<60 THEN 30
80 X=X+1 : Y=0
90 IF X<24 THEN 30 ELSE X=0 : GOTO 30

10 REM ..... 2.17 G Y A K O R L A T
20 INPUT"HANY ORA ";X : INPUT"HANY PERC ";Y
30 GOTO 70
40 CLS
50 PRINTX;" ORA",Y;" PERC"
60 FOR K=1 TO 200 : NEXT
70 Y=Y+1
80 IF Y<60 THEN 40
90 X=X+1 : Y=0
100 IF X>=24 THEN X=0
110 GOTO 40

10 REM ..... 2.18 G Y A K O R L A T
20 FOR I=1 TO 15 : PRINT : NEXT I
30 PRINT"    ------------------------------------------------"
40 FOR I=1 TO 1900 : NEXT
50 GOTO 20

10 REM ..... 2.19 G Y A K O R L A T
20 CLS
30 K=1 : X=0 : L=1
40 FOR I=L TO L+K-1
50   IF X+I>120 THEN STOP
60   SET(X+I,25) : SET(X+I,35) : SET(X+I,45)
70 NEXT I
80 L=L+K : K=K+K
90 FOR F=1 TO 100 : NEXT F
100 GOTO 40
```

```
10 REM ..... 2.20 G Y A K O R L A T
20 CLS
30 FOR I=1 TO 50
40   PRINT"*";
50 NEXT I
```

```
10 REM ..... 2.21 G Y A K O R L A T
20 CLS
30 INPUT"KATI SZAMA ";X
40 CLS
50 INPUT"FERI TIPPJE ";Y
60 PRINT : PRINT"KATI SZAMA ";
70 IF X=Y THEN PRINT"IS UGYANENNYI !" : STOP
80 IF X<Y THEN PRINT"KISEBB" ELSE PRINT"NAGYOBB"
90 PRINT
100 FOR I=1 TO 300 : NEXT
110 GOTO 40
```

```
10 REM ..... 2.22 G Y A K O R L A T
20 CLS
30 X=0 : Y=0 : I=1
40 SET(X,Y)
50 I=I+I
60 FOR L=1 TO 600 : NEXT
70 FOR J=1 TO I
80   X=X+1
90   IF X<128 THEN SET(X,Y) ELSE X=0 : Y=Y+1 :
     IF Y<47 THEN SET(X,Y) ELSE 120
100 NEXT J
110 GOTO 50
120 GOTO 120
```

```
10 REM ..... 2.23 G Y A K O R L A T
20 CLS
30 X=1
40 FOR I=1 TO X : PRINT : NEXT I
50 PRINT"*          *          * *             *"
60 FOR I=1 TO 300 : NEXT :
   X=X+1 :
   IF X>13 THEN X=1
70 X=X+1
80 IF X>13 THEN X=1
90 CLS
100 GOTO 40
```

```
10 REM ..... 2.24 G Y A K O R L A T
20 INPUTX,Y
30 CLS
40 I=4
50 IF I=1 THEN X=X+1 : Y=Y+1
60 IF I=2 THEN X=X-1 : Y=Y-1
70 IF I=3 THEN X=X-1 : Y=Y+1
80 IF I=4 THEN X=X+1 : Y=Y-1
90  IF X<0 THEN IF I=2 THEN I=4 : GOTO 50 ELSE I=1 : GOTO 50
100 IF X>127 THEN IF I=1 THEN I=3 : GOTO 50 ELSE I=2 : GOTO 50
110 IF Y<0 THEN IF I=4 THEN I=1 : GOTO 50 ELSE I=3 : GOTO 50
120 IF Y>47 THEN IF I=3 THEN I=2 : GOTO 50 ELSE I=4 : GOTO 50
130 SET(X,Y)
140 GOTO 50
```

Harmadik fejezet

```
10 REM ..... 3.1 G Y A K O R L A T
20 FOR X=0 TO 127
30   FOR Y=0 TO 47 : SET(X,Y) : NEXT Y
40 NEXT X
50 CLS
60 FOR Y=0 TO 47
70   FOR X=0 TO 127 : SET(X,Y) : NEXT X
80 NEXT Y
90 CLS
100 FOR X=0 TO 127
110   FOR Y=47 TO 0 STEP -1 : SET(X,Y) : NEXT Y
120 NEXT X
130 CLS
140 GOTO 20
```

```
10 REM ..... 3.2 G Y A K O R L A T
20 CLS
30 FOR I=10 TO 20
40   FOR J=5 TO 26 : SET(I,J) : NEXT J
50 NEXT I
60 FOR I=20 TO 40
70   FOR J=1 TO 3 : SET(I,J+4) : SET(I,J+13) : NEXT J
80 NEXT I
90  FOR I=60 TO 70
100    FOR J=5 TO 26 : SET(I,J) : NEXT J
110 NEXT I
120 FOR I=70 TO 90
130   FOR J=1 TO 3
140     FOR K=1 TO 18 : NEXT K
150     SET(I,J+4) : SET(I,J+13) : SET(I,J+23)
160 NEXT J,I
170 GOTO 170
```

```
10 REM ..... 3.3 G Y A K O R L A T
20 CLS
30 DEFINT A-Z
40 FOR Y=0 TO 42 : SET(Y+Y+Y,23) : SET(63,Y) : NEXT Y
50 FOR Y=43 TO 46 : SET(63,Y) : NEXT Y
60 FOR X=0 TO 62
70   FOR Y=0 TO 22 : SET(X,Y) : SET(X+64,Y+24) : NEXT Y
80 NEXT X
90 GOTO 90
```

```
10 REM ..... 3.4 G Y A K O R L A T
20 CLS
30 DEFINT I
40 FOR I=1 TO 5
50   PRINT"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!";
     "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!";
     "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!";
     "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!";
60 NEXT I
70 GOTO 70
```

```
10 REM ..... 3.5 G Y A K O R L A T
20 CLS
30 DEFINT X,Y
40 FOR X=0 TO 127 : SET(X,0) : SET(X,47) : NEXT X
50 FOR Y=0 TO 47
60   SET(0,Y) : SET(1,Y) : SET(126,Y) : SET(127,Y)
70 NEXT Y
80 GOTO 80
```

```
10 REM ..... 3.6 G Y A K O R L A T
20 CLS
30 FOR X=0 TO 127
40   SET(X,20)
50   FOR Y=127 TO X STEP -3 : NEXT Y
60   RESET(X,20)
70 NEXT X
80  FOR X=127 TO 0 STEP-1
90    SET(X,20)
100   FOR Y=127 TO X STEP -3 : NEXT Y
110   RESET(X,20)
120 NEXT X
130 GOTO 20
```

```
10 REM ..... 3.7 G Y A K O R L A T
20 I=100 : X=10 : Y=1 : A=1 : B=57 : S=1
30 CLS
40 FOR K=A TO B STEP S
50   IF K<20 THEN X=X+S : GOTO 90
60   IF K<30 THEN Y=Y+S : GOTO 90
70   IF K<48 THEN X=X-S : GOTO 90
80   Y=Y-S
90   SET(X,Y)
100   FOR J=1 TO I : NEXT J
110   RESET(X,Y)
120 NEXT K
130 IF I=100 THEN I=1 : A=57 : B=1 : S=-1 : GOTO 40 ELSE 20
```

```
10 REM ..... 3.8 G Y A K O R L A T
20 DEFINT X,Y
30 X1=0 : X2=47
40 CLS
50 FOR X=X1 TO X2 STEP 3 : SET(40+X,X1) : SET(40+X2,X) : NEXT X
60 X1=X1+2
70 FOR X=X2 TO X1 STEP -3 : SET(40+X,X2) : SET(40+X1,X) : NEXT X
80 X2=X2-2
90 IF X2>24 THEN 50
100 FOR I=1 TO 50 : NEXT
110 GOTO 30
```

```
10 REM ..... 3.9 G Y A K O R L A T
20 X1=0 : X2=47
30 CLS
40 FOR X=X1 TO X2 : SET(40+X,X1) : NEXT X
50 X1=X1+2
60 FOR X=X2 TO X1 STEP -1 : SET(40+X,X2) : NEXT X
70 X2=X2-2
80 IF X2>24 THEN 40
90 X1=0 : X2=47
100 FOR X=X1 TO X2 : SET(40+X2,X) : NEXT X
110 X1=X1+2
120 FOR X=X2 TO X1 STEP -1 : SET(40+X1,X) : NEXT X
130 X2=X2-2
140 IF X2>24 THEN 100
150 FOR I=1 TO 50 : NEXT
160 GOTO 20
```

```
10 REM ..... 3.10 G Y A K O R L A T
20 K=1 : X=60 : Y=21
30 CLS
40 FOR I=1 TO K
50   IF Y+I>47 THEN 120
60   SET(X+I,Y) : SET(X+K,Y+I)
70 NEXT I
80 X=X+K : Y=Y+K : K=K+K
90 FOR I=1 TO K : SET(X-I,Y) : SET(X-K,Y-I) : NEXT I
100 X=X-K : Y=Y-K : K=K+K
110 GOTO 40
120 FOR X=85 TO 103 : SET(X,0) : NEXT X
130 GOTO 130
```

```
10 REM ..... 3.11 G Y A K O R L A T
20 CLS
30 FOR Y=1 TO 10
40   FOR X=1 TO Y+Y : SET(62+X,Y+3) : SET(63-X,Y+3) : NEXT X
50 NEXT Y
60 FOR Y=14 TO 24
70   FOR X=49 TO 75 : SET(X,Y) : NEXT X
80 NEXT Y
90  FOR Y=5 TO 10
100   FOR X=72 TO 75 : SET(X,Y) : NEXT X
110 NEXT Y
120 FOR Y=16 TO 20
130   FOR X=65 TO 70 : RESET(X,Y) : NEXT X
140 NEXT Y
150 GOTO 150
```

```
10 REM ..... 3.12 G Y A K O R L A T
20 DEFINT A-Z
30 FOR K=1 TO 48
40   CLS
50   FOR X=1 TO K+K
60     FOR Y=0 TO K-1 : SET(X,Y) : NEXT Y
70   NEXT X
80   FOR J=1 TO 200 : NEXT J
90 NEXT K
100 GOTO 100
```

```
10 REM ..... 3.13 G Y A K O R L A T
20 DEFINT A-Z
30 CLS
40 FOR X=1 TO 20
50   SET(X+30,10) : SET(X+30,30) : SET(30,10+X) : SET(50,10+X)
60 NEXT X
70 SET(30,10)
80  FOR Y=11 TO 29
90    FOR J=1 TO 10 : NEXT
100   FOR X=31 TO 49 : SET(X,Y) : NEXT X
110 NEXT Y
120 FOR Y=29 TO 11 STEP -1
130   FOR J=1 TO 100 : NEXT
140   FOR X=49 TO 31 STEP -1 : RESET(X,Y) : NEXT X
150 NEXT Y
160 FOR J=1 TO 200 : NEXT
170 GOTO 80
```

```
10 REM ..... 3.14 G Y A K O R L A T
20 DEFINT A-Z
30 CLS
40 FOR X=1 TO 30
50   FOR Y=1 TO 20 : SET(X+30,Y) : NEXT Y
60 NEXT X
70  FOR Y=1 TO 20
80    FOR X=1 TO 30
90      RESET(X+30,Y)
100     IF X<16 THEN SET(X,Y) ELSE SET(X+60,Y)
110 NEXT X,Y
120 FOR Y=1 TO 20
130   FOR X=30 TO 1 STEP -1
140     SET(X+30,Y)
150     IF X<16 THEN RESET(X,Y) ELSE RESET(X+60,Y)
160 NEXT X,Y
170 GOTO 70
```

```
10 REM ..... 3.15 G Y A K O R L A T
20 CLS
30 FOR X=1 TO 30
40   SET(X,1) : SET(X,30) : SET(X,X) : SET(31-X,X)
50 NEXT X
60 FOR Y=14 TO 1 STEP-1
70   FOR X=Y+1 TO 30-Y
80     SET(X,Y)
90 NEXT X,Y
100 FOR Y=2 TO 14
110   FOR X=Y+1 TO 30-Y
120     FOR J=0 TO 100 : NEXT
130     RESET(X,Y) : SET(X,31-Y)
140 NEXT X,Y
150 FOR J=1 TO 200 : NEXT
160 GOTO 20
```

```
10 REM ..... 3.16 G Y A K O R L A T
20 DEFINT A-Z
30 CLS
40 FOR Y=0 TO 10
50   FOR X=50 TO 70 : SET(X,Y) : RESET(X,40-Y) : NEXT X
60 NEXT Y
70 FOR X=0 TO 127 : SET(X,20) : NEXT X
80  FOR Y=40 TO 30 STEP -1
90    FOR X=50 TO 70 : SET(X,Y) : RESET(X,40-Y) : NEXT X
100 NEXT Y
120 FOR K=1 TO 1500 : NEXT
130 GOTO 40
```

```
10 REM ..... 3.17 G Y A K O R L A T
20 FOR K=0 TO 20
30   CLS
40   SET(63+K+K+K,24+K) : SET(64+K+K+K,24+K) :
     SET(61-K-K-K,23-K) : SET(62-K-K-K,23-K) :
     SET(63+K+K+K,23-K) : SET(64+K+K+K,23-K) :
     SET(61-K-K-K,24+K) : SET(62-K-K-K,24+K)
50   FOR J=1 TO 50 : NEXT
60 NEXT K
70 GOTO 20
```

```
10 REM ..... 3.18 G Y A K O R L A T
20 CLS
30 FOR Y=10 TO 30
40   FOR X=10 TO 60 : SET(X,Y) : NEXT X
50 NEXT Y
60 FOR Y=10 TO 30
70   FOR X=10 TO 60 : RESET(X,Y) : NEXT X
80   FOR X=40-Y TO 90-Y : SET(X,Y) : NEXT X
90 NEXT Y
100 GOTO 100
```

```
10 REM ..... 3.19 G Y A K O R L A T
20 CLS
30 FOR K=0 TO 42
40   FOR J=0 TO 42 : SET(J+J+J,K) : SET(K+K+K,J) : NEXT J
50   FOR J=0 TO 42 : RESET(J+J+J,K) : RESET(K+K+K,J) : NEXT J
60 NEXT K
70 GOTO 30
```

```
10 REM ..... 3.20 G Y A K O R L A T
20 CLS
30 FOR X=0 TO 127
40   FOR Y=10 TO 40 STEP 10 : SET(X,Y) : NEXT Y
60 NEXT X
70 FOR X=0 TO 40 STEP 10
80   FOR Y=0 TO 47
90     IF POINT(X,Y) THEN RESET(X,Y) ELSE SET(X,Y)
100 NEXT Y,X
110 GOTO 110
```

```
10 REM ..... 3.21 G Y A K O R L A T
20 DEFINT A-Z
30 CLS
40 FOR X=1 TO 120
50   FOR Y=10 TO 40 : SET(X,Y) : NEXT Y
60 NEXT X
70 FOR X=50 TO 80
80   FOR Y=0 TO 47
90     IF POINT(X,Y) THEN RESET(X,Y) ELSE SET(X,Y)
100 NEXT Y,X
110 FOR Y=15 TO 35
120   FOR X=45 TO 105
130     IF POINT(X,Y) THEN RESET(X,Y) ELSE SET(X,Y)
140 NEXT X,Y
150 FOR X=15 TO 127
160   IF POINT(X,22) THEN RESET(X,22) ELSE SET(X,22)
170 NEXT X
180 FOR Y=9 TO 47
190   IF POINT(63,Y) THEN RESET(63,Y) ELSE SET(63,Y)
200 NEXT Y
210 GOTO 210
```

```
10 REM ..... 3.22 G Y A K O R L A T
20 CLS
30 FOR K=1 TO 100
40   FOR X=0 TO 100 STEP K
50     IF POINT(X,22) THEN RESET(X,22) ELSE SET(X,22)
60 NEXT X,K
70 FOR X= 0 TO 100
80   IF POINT(X,22) THEN PRINT X;
90 NEXT X
100 GOTO 100
```

```
10 REM ..... 3.23 G Y A K O R L A T
20 CLS
30 FOR X=6 TO 93
40   FOR Y=7 TO 13
50     IF X<15 OR 81<X THEN SET(X,Y)
60 NEXT Y,X
70 GOTO 70
```

Negyedik fejezet

```
10 REM ..... 4.1 G Y A K O R L A T
20 CLS
30 FOR X=0 TO 11 :
      FOR Y=0 TO 6 : SET(X*X,Y*Y) : NEXT Y :
   NEXT X
40 GOTO 40
```

```
10 REM ..... 4.2 G Y A K O R L A T
20 DEFINT A-Z
30 CLS :
     INPUT"A LOTTOSZAMOK ";A,B,C,D,E
40 CLS :
     PRINT A;B;C;D;E
50 FOR I=0 TO 6 :
      FOR J=25 TO 100 : SET(J,5*I+10) : NEXT J :
   NEXT I
60 FOR I=0 TO 30 :
      FOR J=25 TO 100 STEP 5 : SET(J,I+10) : NEXT J :
   NEXT I
70 X=A/15 : Y=A-X*15 : IF Y=0 THEN X=X-1 : Y=15
80 SET(Y*5+22,X*5+12)
90 X=B/15 : Y=B-X*15 : IF Y=0 THEN X=X-1 : Y=15
100 SET(Y*5+22,X*5+12)
110 X=C/15 : Y=C-X*15 : IF Y=0 THEN X=X-1 : Y=15
120 SET(Y*5+22,X*5+12)
130 X=D/15 : Y=D-X*15 : IF Y=0 THEN X=X-1 : Y=15
140 SET(Y*5+22,X*5+12)
150 X=E/15 : Y=E-X*15 : IF Y=0 THEN X=X-1 : Y=15
160 SET(Y*5+22,X*5+12)
170 GOTO 170
```

```
10 REM ..... 4.3 G Y A K O R L A T
20 K=5
30 CLS :
   FOR I=1 TO 2*K :
      FOR J=1 TO K : SET(I,J+6) : NEXT J :
   NEXT I
40 INPUT L,N :
   IF L=0 THEN K=K/N ELSE K=K*N
50 GOTO 30
```

```
10 REM ..... 4.4 G Y A K O R L A T
20 DEFINT A-Z :
   A=0
30 CLS :
   PRINT
40  C=1 :
    FOR L=0 TO 3 :
      FOR K=0 TO 3 :
        IF A=C THEN M=L : N=K : PRINT"     L     "; ELSE PRINT"           ";
50      C=C+1 :
      NEXT K :
      PRINT : PRINT : PRINT :
    NEXT L :
    PRINT
60  C=1
70  FOR L=0 TO 3 : C=C-1 :
      FOR K=0 TO 3 :
        B=C/2
80      IF C=2*B THEN 110
90      IF M=L AND N=K THEN
        FOR X=K*18 TO K*18+17 : SET(X,L*9) : SET(X,L*9+8) : NEXT X :
        FOR Y=L*9 TO L*9+8  : SET(K*18,Y) : SET(K*18+17,Y) : NEXT Y :
        GOTO 110
100     FOR X=K*18 TO K*18+17 :
          FOR Y=L*9 TO L*9+8 : SET(X,Y) : NEXT Y :
        NEXT Y
110     C=C+1 :
    NEXT K,L :
    INPUT A :
    GOTO 30
```

```
10 REM ..... 4.5 G Y A K O R L A T
20 DEFINT X :
   CLS
30 INPUT"A SZAM ";X
40 IF X>32225[(1/5) THEN PRINT"TUL NAGY A SZAM !" : GOTO 30
50 PRINT" X","-X","10*X","X*X": PRINTX,-X,10*X,X*X
60 PRINT : PRINT"X[5=    ";X[5 : PRINT : PRINT
70 PRINT : PRINT :
   GOTO 30
```

```
10 REM ..... 4.6 G Y A K O R L A T
20 CLS :
   INPUT"A HAROM SZAM ";X,Y,Z :
   CLS
30 PRINT"X+Y+Z=";X+Y+Z,"X*Y*Z=";X*Y*Z
40 PRINT"X-Y-Z=";X-Y-Z,"X/Y/Z=";X/Y/Z
50 PRINT"X+Y*Z=";X+Y*Z,"X*Y+Z=";X*Y+Z
60 PRINT"X-Y*Z=";X-Y*Z,"X*Y-Z=";X*Y-Z
70 PRINT"X*Y/Z=";X*Y/Z,"X/Y*Z=";X/Y*Z
80 PRINT"X/Y+Z=";X/Y+Z,"X+Y/Z=";X+Y/Z
90 PRINT"X-Y+Z=";X-Y+Z,"X+Y-Z=";X+Y-Z
100 PRINT"X-Y/Z=";X-Y/Z,"X/Y-Z=";X/Y-Z
110 GOTO 110
```

```
10 REM ..... 4.7 G Y A K O R L A T
20 CLS :
   INPUT N :
   P=0 : K=0 :
   CLS :
   PRINT N : PRINT
30 FOR I=1 TO N : P=P+2*I-1 : K=K+I[3 : NEXT I
40 PRINT"A PARATLAN SZAMOK OSSZEGE : ";P :
   PRINT"A KOBSZAMOK OSSZEGE : ";K
50 GOTO 50
```

```
10 REM ..... 4.8 G Y A K O R L A T
20 CLS :
   INPUT"N= ";N :
   K=0 :
   PRINT"KEREM A SZAMOKAT !"
30 FOR I=1 TO N : INPUT X : K=K+X : NEXT I
40 PRINT"AZ ATLAG = ";K/N
50 STOP

10 REM ..... 4.9 G Y A K O R L A T
20 CLS :
   INPUT"A KOCKA  ELHOSSZA ";X :
   CLS :
   PRINT : PRINT
30 PRINT"TERFOGAT     =";X[3 : PRINT :
   PRINT"FELSZIN      =";6*X*X : PRINT :
   PRINT"ELHOSSZ      =";12*X
40 PRINT : PRINT :
   STOP

10 REM ..... 4.10 G Y A K O R L A T
20 CLS :
   PRINT"A POLINOM  : X -> X * X - 2 * X "
30 PRINT :
   INPUT"MELYIK HELYEN KERED A HELYETTESITESI ERTEKET ";X
40 PRINT : PRINT "X= ";X,"X * X - 2 * X = ";X*X-2*X
50 PRINT : PRINT : PRINT"UJ ERTEK (1)     /KULONBEN VEGE/"; :
   INPUT I
60 IF I=1 THEN 20 ELSE STOP

10 REM ..... 4.11 G Y A K O R L A T
20 L=10000 : S=50000 : V=10000
30 CLS :
   PRINT"UZEMANYAG                  :";L :
   PRINT"FOLDTOL VALO TAVOLSAG   :";S :
   PRINT"SEBESSEG                   :";V : PRINT : PRINT
40 INPUT"UJ SEBESSEG ";V : INPUT"IDO ";T
50 S=S-V*T : L=L-V*V*T/600
60 IF S>10 THEN IF L>0 THEN 30 ELSE
   PRINT"ELFOGYOTT AZ UZEMANYAGOD !" :
   FOR I=1 TO 400 : NEXT : GOTO 80
70 IF S>-10 THEN PRINT"SZERENCSESEN FOLDET ERTEL !" : STOP
80 PRINT : PRINT : PRINT"BUMM !!!!!" : PRINT :
   PRINT"A FOLDNEK UTKOZTEL !"
90 STOP

10 REM ..... 4.12 G Y A K O R L A T
20 CLS
30 INPUT"A MASODPERCEK SZAMA ";M :
   PRINT : PRINT"EZ " :
   N=0 : O=0 : P=0
40 N=N+1 : IF M<N*86400 THEN PRINTN-1,"NAP" ELSE 40
50 M=M-(N-1)*86400
60 O=O+1 : IF M<O*3600 THEN PRINTO-1,"ORA" ELSE 60
70 M=M-(O-1)*3600
80 P=P+1 : IF M<P*60 THEN PRINTP-1,"PERC" ELSE 80
90 PRINTM-(P-1)*60,"MASODPERC" : PRINT : PRINT : PRINT :
   GOTO 30
```

```
10 REM ..... 4.13 G Y A K O R L A T
20 DEFINT A-Z :
   CLS
30 INPUT"A KETTES SZAMRENDSZERBELI SZAM ";X :
   J=X
40 A=0 : I=1
50 Z=X/2 : IF X=2*Z THEN X=X/10 ELSE X=(X-1)/10 : A=A+I
60 IF X>0 THEN I=I*2 : GOTO 50
70 PRINTJ,"A TIZES SZAMRENDSZERBEN :";A : PRINT : PRINT
80 INPUT"A TIZES SZAMRENDSZERBELI SZAM ";Y :
   J=Y
90 A=0 : I=1
100 Z=Y/2 : IF Y=2*Z THEN Y=Y/2 ELSE Y=(Y-1)/2 : A=A+I
110 IF Y>0 THEN I=I*10 : GOTO 100
120 PRINTJ,"A KETTES SZAMRENDSZERBEN :";A : PRINT : PRINT : PRINT :
    GOTO 30
```

```
10 REM ..... 4.14 G Y A K O R L A T
20 CLS :
   DEFINT A-Z
30 INPUT"A KET SZAM ";X,Y
40 IF X<Y THEN Z=X : X=Y : Y=Z
50 Z=X/Y : PRINTX;"=";Y;"*";Z;"+";X-Z*Y
60 PRINT : PRINT :
   GOTO 30
```

```
10 REM ..... 4.15 G Y A K O R L A T
20 CLS :
   DEFINT A-Z
30 INPUT"A SZAM  ";X :
   PRINT"AZ ENNEL KISEBB PRIMSZAMOK :" :
   IF X=1 THEN 120
40 PRINT" 1 "; :
     IF X=2 THEN 120
50 PRINT" 2 ";
60 K=3
70 IF K>=X THEN 120
80 FOR I=3 TO K/2+1 STEP 2 :
       Z=K/I :
       IF K=Z*I THEN 110
90 NEXT I
100 PRINT K;
110 K=K+2 :
    GOTO 70
120 PRINT : PRINT : GOTO 30
```

```
10 REM ..... 4.16 G Y A K O R L A T
20 CLS :
   INPUT"AZ OT SZAM ";A,B,C,D,E
30 X=0 : I=0
40 IF X>=A OR X>=B OR X>=C OR X>=D OR X>=E THEN X=X-1 :
     GOTO 40
50 IF A<B THEN Z=B ELSE Z=A
60 IF Z<C THEN Z=C
70 IF Z<D THEN Z=D
80 IF Z<E THEN Z=E
90 PRINTZ; : I=I+1 : IF I>4 THEN STOP
100 IF A=Z THEN A=X
110 IF B=Z THEN B=X
120 IF C=Z THEN C=X
130 IF D=Z THEN D=X
140 IF E=Z THEN E=X
150 GOTO 50
```

```
10 REM ..... 4.17 G Y A K O R L A T
20 CLS :
   DEFINT A-Z
30 INPUT" A KET SZAM ";X,Y : A=X*Y
40 IF X=Y THEN 60
50 IF X>Y THEN X=X-Y : GOTO 40 ELSE Y=Y-X : GOTO 40
60 PRINT" L.N.K.O. :";X : PRINT" L.K.K.T. :";A/X
70 PRINT : PRINT : GOTO 30
```

```
10 REM .... 4.18 G Y A K O R L A T
20 CLS :
   DEFINT A-Z
30 INPUT"A KET SZAM ";X,Y
40 IF X>Y THEN Z=Y : Y=X : X=Z
50 X=X+1 : IF X>=Y THEN 80
60 FOR I=2 TO X/2 :
     Z=X/I :
     IF X=Z*I THEN 50
70 NEXT I :
   PRINT X :
   GOTO 90
80 PRINT"NINCS ILYEN PRIMSZAM !"
90 STOP
```

```
10 REM ..... 4.19 G Y A K O R L A T
20 DEFINT A-Z :
   CLS :
   INPUT"A KET SZAM "; X,Y :
   M=0 : N=0
30 FOR I=2 TO X/2 :
     Z=X/I
40   IF X=Z*I THEN M=M+1
50 NEXT
60 FOR I=2 TO Y/2 :
     Z=Y/I
70   IF Y=Z*I THEN N=N+1
80 NEXT
90 IF M=N THEN PRINT"AZ OSZTOK SZAMA UGYANANNYI  !":GOTO:
100 PRINT"TOBB OSZTOJA VAN A KOVETKEZO SZAMNAK :"
110 IF M<N THEN PRINT Y ELSE PRINT X
120 STOP
```

```
10 REM ..... 4.20 G Y A K O R L A T
20 DEFINT X,Y
30 CLS
40 FOR X=5 TO 45
50   FOR Y=5 TO 45
60     IF X=10 OR X=20 OR X=30 OR X=40 OR 11<X AND X<19
          OR Y=10 OR Y=20 OR Y=30 OR Y=40 OR 11<Y AND Y<19
         THEN 70
         ELSE SET(X,Y)
70 NEXT Y,X
80 GOTO 80
```

Ötödik fejezet

```
10 REM ..... 5.1 G Y A K O R L A T
20 CLS :
   X=63 : Y=23
30 FOR B=0 TO 20 :
       IF B=1 OR B=2 OR B=3 OR B=5 OR B=6 OR B=7 OR B=8 OR B=12
       THEN 40 ELSE A=1.8*SQR(400-B*B) :
       SET(X+A,Y+B) : SET(X+A,Y-B) : SET(X-A,Y+B) : SET(X-A,Y-B)
40 NEXT B
50 GOTO 50
```

```
10 REM ..... 5.2 G Y A K O R L A T
20 CLS :
   X=63 : Y=25
30 PRINT:PRINT:PRINT"                                  12":PRINT:
   PRINT:PRINT:PRINT:PRINT:PRINT"                   ";:
   PRINT"9                              3";:
   PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:
   PRINT"                            6"
40 FOR B=0 TO 20 :
       IF B=1 OR B=2 OR B=3 OR B=5 OR B=6 OR B=7 OR B=8 OR B=12
       THEN 50 ELSE A=1.8*SQR(400-B*B) :
       SET(X+A,Y+B) : SET(X+A,Y-B) : SET(X-A,Y+B) : SET(X-A,Y-B)
50 NEXT
60 FOR I=0 TO 15 : SET(63,25-I) : NEXT I :
   FOR I=0 TO 100 : NEXT :
   FOR I=15 TO 0 STEP -1 : RESET(63,25-I) : NEXT I
70 FOR I=0 TO 30 : SET(63+I,25) : NEXT I :
   FOR I=0 TO 100 : NEXT :
   FOR I=30 TO 0 STEP -1 : RESET(63+I,25) : NEXT I
80 FOR I=0 TO 15 : SET(63,25+I) : NEXT I :
   FOR I=0 TO 100 : NEXT :
   FOR I=15 TO 0 STEP -1 : RESET(63,25+I) : NEXT I
90 FOR I=0 TO 30 : SET(63-I,25) : NEXT I :
   FOR I=0 TO 100 : NEXT :
   FOR I=30 TO 0 STEP -1 : RESET(63-I,25) : NEXT I
100 GOTO 60
```

```
10 REM ..... 5.3 G Y A K O R L A T
20 CLS :
   I=1
30 X=63+EXP(-.01*I)*75*SIN(I*3.14/32) :
   SET(X,23) :
   FOR K=1 TO 50 : NEXT :
   RESET(X,23) :
   I=I+1 :
   GOTO 30
```

```
10 REM ..... 5.4 G Y A K O R L A T
20 CLS :
   FOR I=1 TO 15 : PRINT : NEXT I :
   I=1
30 X=EXP(-.01*I)*SIN(I*3.14/32)*75+63 :
   SET(X,47) :
   I=I+1 :
   PRINT :
   GOTO 30
```

```
10 REM ..... 5.5 G Y A K O R L A T
20 DEFINTA-Z :
   CLS
30 I=0 :
   INPUT"A SZAM ";X  :
   IF ABS(X)>9999 THEN 30
40 IF X<0 THEN X=-X : I=1
50 K=0 : PRINT
60 K=K+1 : A=10[K : IF X<A THEN 70 ELSE 60
70 IF I=1 THEN 90
80 PRINTX,"=",X/10[(K-1);"E+";(K-1) : GOTO 100
90 PRINT-X,"=",-X/10[(K-1);"E+";(K-1)
100 PRINT : PRINT : PRINT :
    GOTO 30

10 REM ..... 5.6 G Y A K O R L A T
20 CLS
30 PRINT" AZ EGYENLET :   S I N ( X ) = X / 2" : PRINT
40 INPUT"X= ";X : PRINT
50 PRINT"X= ";X,"SIN(X)-X/2= ";SIN(X)-X/2 : PRINT : PRINT
60 IF ABS(SIN(X)-X/2)<.0001 THEN PRINT" A GYOK : ";X ELSE 40
70 PRINT
80 STOP

10 REM ..... 5.7 G Y A K O R L A T
20 CLS
30 INPUT"AZ OLDAL A=";A :
   IF A>0 THEN 40 ELSE PRINT"
   EZ NEM EGY HAROMSZOG OLDALA !" : GOTO 30
40 INPUT"A KET SZOG (RADIANBAN MERVE !) X,Y=";X,Y
50 Z=3.14159-X-Y :
   IF Z<0 OR X<0 OR Y<0 THEN PRINT"
   EZEK NEM EGY HAROMSZOG SZOGEI !" : GOTO 40
60 PRINT : PRINT"A HARMADIK SZOG Z=";Z
70 PRINT : PRINT"A MASIK KET OLDAL B=";A*SIN(X)/SIN(Z),
    "C=";A*SIN(Y)/SIN(Z)
80 PRINT : PRINT : PRINT : GOTO 30

10 REM .....  5.8 G Y A K O R L A T
20 CLS :
   PRINT"A KOSZINUSZ FUGGVENY HELYETTESITESI ERTEKEI :"
30 INPUT"KEZDOERTEK ";X : INPUT"LEPESKOZ ";Y
40 FOR I=X TO X+10*Y STEP Y : PRINT I,COS(I) : NEXT I
50 PRINT : PRINT"TOVABB (1)       VEGE (0)";
60 INPUT I : IF I=1 THEN 20 ELSE IF I=0 THEN STOP ELSE 60

10 REM ..... 5.9 G Y A K O R L A T
20 CLS
30 INPUT"N= ";N
40 PRINT" N                  (1+1/N)[N         (1+1/N)[(N+1)"
50 FOR I=N TO N+9 : PRINTI,(1+1/I)[I,(1+1/I)[(I+1) : NEXT I
60 PRINT : PRINT : PRINT"UJ N ERTEK (1)";
70 INPUTI : IF I=1 THEN 30
80 STOP
```

Hatodik fejezet

```
10 REM ..... 6.1 G Y A K O R L A T
20 CLS :
   A=RND(127) : B=RND(47) : M=RND(127) : N=RND(47) :
   X=RND(127) : Y=RND(47)
30 SET(A,B) : SET(M,N) : SET(X,Y) :
   IF INKEY$="" THEN CLS : GOTO 30 ELSE 20
```

```
10 REM ..... 6.2 G Y A K O R L A T
20 CLS
30 GOTO 50
40 CLS :
   X=RND(127) : Y=RND(47) : SET(X,Y)
50 B$=INKEY$ : IF B$="" THEN 50 ELSE IF A$<>B$ THEN A$=B$ : GOTO 40
60 FOR I=1 TO 50 : NEXT :
   RESET(X,Y) :
   FOR I=1 TO 50 : NEXT
70 A$=INKEY$ : IF A$="" OR A$=B$ THEN SET(X,Y) : GOTO 60 ELSE 40
```

```
10 REM ..... 6.3 G Y A K O R L A T
20 DEFINT I :
   CLS :
   X=63 : Y=23
30 I=RND(4)
40 IF I=1 THEN X=X+4 : IF X>127 THEN X=127
50 IF I=2 THEN Y=Y+2 : IF Y>47 THEN Y=47
60 IF I=3 THEN X=X-4 : IF X<0 THEN X=0
70 IF I=4 THEN Y=Y-2 : IF Y<0 THEN Y=0
80 RESET(A,B) : SET(X,Y) :
   A=X : B=Y :
   GOTO 30
```

```
10 REM ..... 6.4 G Y A K O R L A T
20 CLS :
   SET(1,1) : SET(1,20) : SET(25,1) : SET(25,27) : SET(50,5) : SET(50,15)
30 I=RND(6)
40 IF I=1 THEN X=1 : Y=1 ELSE IF I=2 THEN X=1 : Y=20 ELSE
   IF I=3 THEN X=25 : Y=1 ELSE IF I=4 THEN X=25 : Y=27 ELSE
   IF I=5 THEN X=50 : Y=5 ELSE IF I=6 THEN X=50 : Y=15
50 FOR I=1 TO 5 :
     RESET(X,Y) : FOR J=1 TO 50 : NEXT J : SET(X,Y) :
   NEXT I
60 GOTO 30
```

```
10 REM ..... 6.5 G Y A K O R L A T
20 IF INKEY$="" THEN 20 ELSE CLS
30 X=RND(120)+3 : Y=RND(40)+3 : SET(X,Y) :
   FOR I=1 TO 3 : SET(X+I,Y+I) : SET(X-I,Y+I) : NEXT I :
   FOR I=-3 TO 3 STEP 2 : SET(X+I,Y+3) : NEXT I
40 GOTO 20
```

```
10 REM ..... 6.6 G Y A K O R L A T
20 FOR I=1 TO 15 : PRINT : NEXT :
   PRINT" ----------------------------------------------"
30 I=RND(800)+20 :
   FOR J=1 TO I : NEXT :
   GOTO 20
```

```
10 REM ..... 6.7 G Y A K O R L A T
20 DEFINT A-Z :
   CLS :
   PRINT : PRINT : PRINT
30 PRINT"        +++++++     +++++++" :
   FOR I=1 TO 7 :
   PRINT"        +      +     +      +" :
   NEXT :
   PRINT"        +++++++     +++++++"
40 I=RND(2) : J=RND(2)
50 FOR B=3 TO 23 :
     FOR A=1 TO 10 :
        IF J=1 THEN SET(13+(I-1)*22+A,9+B)
               ELSE RESET(13+(I-1)*22+A,35-B)
60 NEXT A,B
70 GOTO 40
```

```
10 REM ..... 6.8 G Y A K O R L A T
20 DEFINTA-Z :
   A=0 : B=0 : C=0 : D=0 : E=0 :F=0 :
   CLS
30 PRINT" DOBALOM A KOCKAT !!!     A DOBASOK SZAMA :"
40 FOR K=1 TO 1000 :
     I=RND(6) :
     PRINTK; :
     IFI=1THENA=A+1ELSEIFI=2THENB=B+1ELSEIFI=3THENC=C+1
     ELSEIFI=4THEND=D+1ELSEIFI=5THENE=E+1ELSEF=F+1
50 NEXT K
60 CLS :
   PRINT"A KOCKADOBASOK ELOSZLASA :" : PRINT : PRINT"1-es",A :
   PRINT"2-es",B : PRINT"3-as",C : PRINT"4-es",D : PRINT"5-os",E :
   PRINT"6-os",F
70 STOP
```

```
10 REM ..... 6.9 G Y A K O R L A T
20 CLS :
   INPUT"A KEZDOPONT KOORDINATAI ";X,Y
30 CLS :
   SET(X,Y)
40 B$=A$
50 IF B$="F" THEN Y=Y-1 : IF Y<0 THEN Y=0
60 IF B$="L" THEN Y=Y+1 : IF Y>47 THEN Y=47
70 IF B$="J" THEN X=X+1 : IF X>127 THEN X=127
80 IF B$="B" THEN X=X-1 : IF X<0 THEN X=0
90 SET(X,Y)
100 A$=INKEY$ : IF A$="" THEN 50 ELSE 40
```

```
10 REM ..... 6.10 G Y A K O R L A T
20 DEFINT X
30 CLS :
   X=RND(4)
40 IF X=1 THEN PRINT"      TAVASZ" ELSE IF X=2 THEN PRINT"
   NYAR" ELSE IF X=3 THEN PRINT"                              OSZ"
   ELSE PRINT"

                               TEL"
50 FOR I=1 TO 200 : NEXT : GOTO 30
```

```
10 REM ..... 6.11 G Y A K O R L A T
20 CLS
30 X=RND(100) : Y=RND(100) : PRINT : PRINT
40 PRINT"MENNYIVEL EGYENLO ?" : PRINT :
   PRINTX;"  *  ";Y;"  =  "; : INPUT A : PRINT : PRINT
50 IF X*Y=A THEN PRINT" JOL SZAMOLTAL ! " : PRINT : PRINT :
   GOTO 30
60 PRINT" ELLENORIZD !!!" : GOTO 40
```

```
10 REM ..... 6.12 G Y A K O R L A T
20 CLS
30 PRINT : PRINT : PRINT :
   X=RND(100) : Y=RND(100)
40 PRINT : PRINT" MI AZ EREDMENY ?",X*Y;"  :  ";X;"  =  ";
50 INPUT A : IF Y=A THEN PRINT"JOL SZAMOLTAL !" : PRINT : GOTO 30
60 PRINT"ELLENORIZD !!!" : PRINT : GOTO 40
```

```
10 REM ..... 6.13 G Y A K O R L A T
20 CLS :
   PRINT : PRINT : PRINT :
   PRINT"HA MEGJELENIK EGY FELKIALTOJEL A KEPERNYON,
   NYOMJAL MEG EGY BILLENTYUT !"
30 PRINT : PRINT : PRINT : PRINT"K E Z D H E T J U K ?"
40 IF INKEY$="" THEN 40 ELSE CLS
50 K=RND(1000)+100 :
   FOR I=1 TO K : NEXT :
   I=0 :
   A$=INKEY$ :
   PRINT"        !"
60 IF INKEY$="" THEN I=I+1 : GOTO 60
70 PRINT : PRINT : PRINT :
   PRINT I,"IDOEGYSEGRE VOLT SZUKSEGED !" : PRINT : PRINT :
   PRINT"FOLYTATOD -I        NEM - N"
80 A$=INKEY$ : IF A$="I" THEN 20 ELSE IF A$="N" THEN STOP ELSE 80
```

```
10 REM ..... 6.14 G Y A K O R L A T
20 CLS :
   PRINT"HA MEGJELENIK AZ  1-ES SZAM, AKKOR AZ 1-ES,
            HA A  9-ES SZAM, AKKOR  A 9-ES BILLENTYUT KELL
                                            LENYOMNI  !
     K E Z D H E T J U K ?"
30 IF INKEY$="" THEN 30 ELSE CLS
40 I=RND(1000)+100 :
   FOR K=1 TO I : NEXT :
   I=0 :
   A$=INKEY$
50 IF RND(2)-1 THEN 80
60 PRINT"1"
70 IF INKEY$="1" THEN 100 ELSE I=I+1 : GOTO 70
80 PRINT"    9"
90 IF INKEY$="9" THEN 100 ELSE I=I+1 : GOTO 90
100 PRINT : PRINT : PRINT"AZ IDOEGYSEG : ";I : PRINT :
    PRINT : PRINT : PRINT"ELOVASTAD ?"
110 IF INKEY$="" THEN 110 ELSE 20
```

```
10 REM .... 6.15 G Y A K O R L A T
20 CLS :
   PRINT : PRINT"A JATEKOSOK A   - D - BILLENTYU
   LENYOMASAVAL DOBHATNAK !
                                 K E Z D H E T J U K ?"
30 IF INKEY$="" THEN 30 ELSE CLS : A=0 : B=0
40 INPUT"    A    B   JATEKOS TIPPJE ";X : PRINT : PRINT :
   PRINT"AZ A JATEKOS DOB !"
50 IF INKEY$="D" THEN Y=RND(6) : PRINT Y  ELSE 50
60 IF X=Y THEN A=A+X ELSE A=A-ABS(X-Y)
70 PRINT"A= ";A : PRINT : PRINT
80 IF A>99 THEN PRINT : PRINT"AZ A JATEKOS NYERT !" : STOP
90 INPUT"   AZ    A   JATEKOS TIPPJE ";X : PRINT : PRINT :
   PRINT"A B JATEKOS DOB !"
100 IF INKEY$="D" THEN Y=RND(6) : PRINT Y ELSE 100
110 IF X=Y THEN B=B+X ELSE B=B-ABS(X-Y)
120 PRINT"B= ";B : PRINT : PRINT
130 IF B>99 THEN PRINT : PRINT"A B JATEKOS NYERT !" ELSE 40
140 STOP
```

Hetedik fejezet

```
10 REM ..... 7.1 G Y A K O R L A T
20 DEFSTR A,B :
   CLS
30 INPUT"A SZO ";A : PRINT : PRINT"A SZO FORDITVA ";
40 X=LEN(A) :
   FOR I=X TO 1 STEP-1 : PRINT MID$(A,I,1); : NEXT I
50 PRINT : PRINT : PRINT : GOTO 30
```

```
10 REM ..... 7.2 G Y A K O R L A T
20 DEFSTR A :
   CLS
30 INPUT"A SZOVEG ";A :
   X=LEN(A) : J=0
40 FOR I=1 TO X :
     IF MID$(A,I,1)="A" THEN J=J+1
50 NEXT I
60 PRINT : PRINT"AZ A BETUK SZAMA :";J : PRINT : PRINT
70 GOTO 30
```

```
10 REM ..... 7.3 G Y A K O R L A T
20 DEFSTR A,B :
   CLS
30 INPUT"A SZO ";A :
   X=LEN(A)
40 FOR I=1 TO X :
     B=MID$(A,I,1) :
     IF B="A" THEN PRINT"*"; ELSE PRINT B;
50 NEXT I
60 PRINT : PRINT : GOTO 30
```

```
10 REM ..... 7.4 G Y A K O R L A T
20 DEFSTR A,B :
   CLS
30 INPUT"A SZOVEG ";A :
   X=LEN(A)
40 FOR I=1 TO X :
     B=MID$(A,I,1) :
     IF B<>" " THEN PRINT B;
50 NEXT I
60 PRINT : PRINT : GOTO 30
```

```
10 REM ..... 7.5 G Y A K O R L A T
20 DEFSTR A,B :
   CLS
30 INPUT"A SZO ";A :
   X=LEN(A) :
   PRINT
40 FOR I=1 TO X : B=MID$(A,I,1) : PRINT CHR$(ASC(B)+32); : NEXT I
50 PRINT : PRINT : PRINT : GOTO 30
```

```
10 REM ..... 7.6 G Y A K O R L A T
20 CLS
30 FOR I=0 TO 14 :
     PRINT RIGHT$(STR$(I*I),LEN(STR$(I*I))-1); :
   NEXT I
40 PRINT : PRINT : STOP
```

```
10 REM ..... 7.7 G Y A K O R L A T
20 CLS :
   DEFSTR K : DEFINT T,Z
30 INPUT"A KETTES SZAMRENDSZERBELI SZAM :";K :
   X=LEN(K) : L=1
40 FOR I=X TO 1 STEP-1 :
     IF MID$(K,I,1)="1" THEN T=T+L
50   L=L*2 :
   NEXT I :
   PRINT : PRINT K,"A TIZES SZAMRENDSZERBEN :",T
60 PRINT : PRINT : INPUT"A TIZES SZAMRENDSZERBELI SZAM ";T
70 K=""  : L=T
80 Z=T/2 : IF T=2*Z THEN T=T/2 : K="0"+K ELSE T=(T-1)/2 : K="1"+K
90 IF T>0 THEN 80 ELSE PRINT : PRINT :
                        PRINT L,"A KETTES SZAMRENDSZERBEN :",K
100 PRINT : PRINT : PRINT : GOTO 30
```

```
10 REM ..... 7.8 G Y A K O R L A T
20 CLS :
   DEFSTR A
30 I=9+RND(90) : J=9+RND(90) : A=STR$(I*J) : X=LEN(A) :
   A=RIGHT$(A,X-1) :
   IF X=4 THEN A="0"+A
40 PRINT A, :
   FOR I=1 TO 200 : NEXT :
   GOTO 30
```

```
10 REM ..... 7.9 G Y A K O R L A T
20 DEFSTR A,B : DEFINT X,Z :
   CLS
30 FOR I=1 TO 10 :
     PRINT I;". SZAM= "; : INPUT A : B=B+A :
   NEXT I
40 X=LEN(B) : Z=X/5 : X=X-5*Z : Z=5*Z-1 :
   CLS
50 FOR I=1 TO Z STEP 5 : PRINT MID$(B,I,5) : NEXT I
60 PRINT
70 STOP
```

```
10 REM ..... 7.10 G Y A K O R L A T
20 DEFSTR A,B :
   CLS
30 INPUT"A MUVELET ";A :
   I=0 : X=LEN(A)
40 I=I+1 : IF I>X THEN CLS : GOTO 30
50 B=MID$(A,I,1)
60 IF B="+" THEN J=1 ELSE IF B="*" THEN J=0 ELSE 40
70 C=VAL(LEFT$(A,I-1)) : D=VAL(RIGHT$(A,X-I))
80 IF J THEN C=C+D ELSE C=C*D
90 CLS :
   PRINT A;" = ";C : PRINT : PRINT : PRINT : GOTO 30
```

Nyolcadik fejezet

```
10 REM ..... 8.1 G Y A K O R L A T
20 CLS
30 PRINT@RND(1016),"VILLOG" :
   FOR I=1 TO 100 : NEXT :
   GOTO 20
```

```
10 REM ..... 8.2 G Y A K O R L A T
20 CLS :
   INPUT"HANYADIK KARAKTERPOZICIOTOL IRJAK ";K :
   INPUT"MIT IRJAK KI ";A$
30 CLS  :
   PRINT@K,A$ :
   FOR I=1 TO 1000 : NEXT :
   GOTO 20
```

```
10 REM ..... 8.3 G Y A K O R L A T
20 CLS :
   PRINT@285," /////" : PRINT@541," ....." :
   PRINT@349,". O O ."; : PRINT@413,".  -  ."; :
   PRINT@477,".  o  .";
30 GOTO 30
```

```
10 REM ..... 8.4 G Y A K O R L A T
20 CLS
30 FOR I=1 TO 28 : READ X,Y : SET(X,Y) : NEXT I
40 GOTO 40
50 DATA 60,10,61,10,62,10,63,10,64,10,59,11,59,12,61,12,63,12,
   59,13,59,14,62,14,59,15,59,16,62,16,59,17,60,18,61,18,62,
   18,63,18,64,18,65,17,65,16,65,15,65,14,65,13,65,12,65,11
```

```
10 REM ..... 8.5 G Y A K O R L A T
20 CLS
30 PRINT@5*64,"    O - C - O","SZENDIOXID"
40 PRINT@10*64,"   H - O - H","VIZ"
50 GOTO 50
```

```
10 REM ..... 8.6 G Y A K O R L A T
20 CLS
30  PRINT"                      +       +     +  +  +" :
    PRINT"           +    + +       +                  +" :
    PRINT"     +                                           +" :
    PRINT"  +                                            +"
40  PRINT"  +                                            +" :
    PRINT"   +                                            +" :
    PRINT"    +                                             +";
    PRINT"      +                                           +"
50  PRINT"      +                                        +" :
    PRINT"       +                                          +";
    PRINT"        +                                         +";
    PRINT"     +                                          +";
60  PRINT"  +                                             +" :
    PRINT" +                     +  +                  +" :
    PRINT"+            +                    +            +" :
    PRINT"+ +  +                               +  +  +   +";
70 FOR I=1 TO 67 : READ X,Y : SET(X,Y) : NEXT I
80 PRINT@6*64+30,"B U D A P E S T";
90 GOTO 90
100 DATA 50,2,50,3,49,4,48,5,48,6,47,7,47,8,46,9,46,10,46,11,
    45,12,45,13,45,14,46,15,46,16,46,17,47,18,47,19,48,20,
    48,21,48,22,48,23,49,24,50,25,50,26,51,27,51,28,52,29,
    51,30,50,31,50,32,49,33,48,34,47,33,46,34,45,35,44,36,
    43,37,42,38
110 DATA 38,39,35,40,32,41,31,42,30,42,29,43,26,44,20,45,15,46,
    10,47
120 DATA 48,34,49,35,51,36,53,37,55,38,54,39,54,40,53,41,
    52,42,51,44,50,46
130 DATA 46,12,47,13,48,14,48,15,48,16,47,17,46,18
```

```
10 REM ..... 8.7 G Y A K O R L A T
20 CLS
30 FOR X=16 TO 122 STEP 3 : SET(X,47-6*LOG(X-15)) : NEXT X
40 FOR X=0 TO 127 : SET(X,36) : NEXT X :
   FOR Y=0 TO 45 : SET(10,Y) : NEXT Y
50 PRINT@13*64+2,"0";: PRINT@13*64+10,"1"; :
   PRINT@35,"L O G A R I T M U S";
60 GOTO 60
```

```
10 REM ..... 8.8 G Y A K O R L A T
20 DEFINT I :
   CLS :
   X=0
30 PRINT" X";TAB(14);"INT(X)";TAB(27);"25*X";TAB(40);
   "X*X";TAB(53);"SQR(X)"; :
   FOR I=0 TO 63 : PRINT@64+I,"-"; : NEXT I
40 PRINT@3*64,""; : INPUT"A KEZDOERTEK= ";X : INPUT"A LEPESKOZ= ";K :
   I=7
50 I=I+1 : IF I=15 THEN 40
60 PRINT@I*64,X; : PRINT TAB(13);INT(X);TAB(26);25*X;TAB(39);
   X*X;TAB(52);SQR(X);
70 IF INKEY$="" THEN 70 ELSE X=X+K : GOTO 50
```

```
10 REM ..... 8.9 G Y A K O R L A T
20 CLS
30 FOR X=1 TO 81 : SET(X,10) : SET(X,22) : NEXT X :
   FOR X=1 TO 81 STEP 20 : FOR Y=11 TO 21 : SET(X,Y) : NEXT Y,X
40 X=RND(4) :
   PRINT@5*64-5+10*X,"!"; :
   FOR I=1 TO 200 : NEXT :
   PRINT@5*64-5+10*X," "; :
   GOTO 40.
```

```
10 REM ..... 8.10 G Y A K O R L A T
20 CLS :
   L=0
30 FOR A=0 TO 80 STEP 10 :
     FOR B=0 TO 40 STEP 5 :
       IF L=-1 THEN L=0 : GOTO 50 ELSE L=-1
40     FOR I=0  TO 9 : FOR K=0 TO 4 : SET(A+I,B+K) : NEXT K,I
50 NEXT B,A
60 A=RND(9)-1 : B=RND(9)-1 : A=A*10 : B=B*5
70 FOR I=1 TO 9 :
     READ X,Y :
     IF POINT(A,B) THEN RESET(A+X,B+Y) ELSE SET(A+X,B+Y)
80 NEXT I :
   FOR J=1 TO 200 : NEXT J
90 IF POINT(A,B) THEN FORI=0TO9:FORK=0TO4:SET(A+I,B+K):NEXT:NEXT
                  ELSE FORI=0TO9:FORK=0TO4:RESET(A+I,B+K):NEXT:NEXT
100 RESTORE :
    GOTO 60
110 DATA 3,1,4,1,5,1,3,2,4,2,2,3,3,3,4,3,5,3
```

```
10 REM ..... 8.11 G Y A K O R L A T
20 CLS  :
   PRINT@5*64+25,"I N D U L H A T ?" :
   I=0 : K=0 :
   PRINT@8*64+20,I,":",K
30 IF INKEY$="I" THEN
   PRINT@5*64+25,"
                  ELSE 30
40 K=K+1 : IF K=60 THEN K=0 : I=I+1
50 PRINT@8*64+20,I,":",K
60 IF INKEY$="S" THEN 70 ELSE 40
70 IF INKEY$="0" THEN 20 ELSE 70
```

```
10 REM ...., 8.12 G Y A K O R L A T
20 CLS
30 FOR I=1 TO 3
40   FOR K=0 TO 15 : READ A : PRINT@K*64+I*15,A[I; : NEXT K
50   RESTORE :
   NEXT I
60 GOTO 60
70 DATA 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
```

```
10 REM ..... 8.13 G Y A K O R L A T
20 CLS :
   FOR I=5  TO 54 :  PRINT@6*64+I, CHR$(K) : NEXT I
30 FOR I=1 TO 200 : NEXT
40 FOR I=54 TO 5 STEP-1 : PRINT@6*64+I,CHR$(32) : NEXT I
50 GOTO 20
```

```
10 REM ..... 8.14 G Y A K O R L A T
20 CLS :
   K=31+RND(160)
30 FOR I=15 TO 4 STEP-1 :
     PRINT@I*64+50-I,CHR$(K); :  PRINT@I*64+18+I,CHR$(K); :
   NEXT I
40 IF INKEY$="" THEN 40 ELSE 20
```

```
10 REM ..... 8.15 G Y A K O R L A T
20 CLS
30 FOR I=15 TO 4 STEP -1 :
     PRINT@I*64+50-I,CHR$(31+RND(160)); :
     PRINT@I*64+18+I,CHR$(31+RND(16)); :
   NEXT I
40 IF INKEY$="" THEN 40 ELSE 30
```

```
10 REM ..... 8.16 G Y A K O R L A T
20 CLS :
   I=RND(160)+31 : A$=CHR$(I)
30 FOR X= 5 TO 9 :
     PRINT@X*64+21,A$; : PRINT@X*64+30,A$; :
     PRINT@4*64+19+X,A$; : PRINT@10*64+19+X,A$; :
   NEXT X
40 PRINT@7*64+24,I;
50 IF INKEY$="" THEN 50 ELSE 20
```

```
10 REM ..... 8.17 G Y A K O R L A T
20 CLS
30 I=RND(8) : J=RND(8) :
   PRINT@(I-1)*128+J*7,CHR$((I-1)*8+J+127); :
   GOTO 30
```

```
10 REM ..... 8.18 G Y A K O R L A T
20 CLS
30 FOR X=1 TO 20 :
     SET(41,X) : SET(83,X) : SET(41+2*X,0) : SET(41+2*X,21) :
   NEXT X
40 PRINT@2*64+25,CHR$(131);CHR$(131);CHR$(131); :
   PRINT@2*64+35,CHR$(131);CHR$(131);CHR$(131);
50 PRINT@3*64+31,CHR$(191);
60 PRINT@5*64+30,CHR$(141);CHR$(140);CHR$(142);
70 GOTO 70
```

```
10 REM .....  8.19 G Y A K O R L A T
20 DEFSTR A,B
30 CLS
40 PRINT@0,""; : INPUT"A SZO ";A
50 X=LEN(A) : I=0
60 I=I+1
70 IF I>X-3 THEN PRINT"NEM JO A SZO !!" : GOTO 100
80 IF MID$(A,I,3)="SSZ" THEN B=LEFT$(A,I)+"Z-" : A=RIGHT$(A,X-I)
                         ELSE 60
90 PRINT@448-LEN(B),B;A
100 FOR K=0 TO 3000 : NEXT : GOTO 30
```

Kilencedik fejezet

```
10 REM ..... 9.1 G Y A K O R L A T
20 DEFSTR A :
   CLS
30 A(0)="V" : A(1)="E" : A(2)="L" : A(3)=A(1) : A(4)="T" :
   A(5)=A(2) : A(6)=A(1) : A(7)="N"
40 X=RND(8)-1 : PRINT@6*64+20+3*X,A(X); :
   FOR I=1 TO 200 : NEXT :
   GOTO 40
```

```
10 REM ..... 9.2 G Y A K O R L A T
20 CLS :
   PRINT"KEREM A NEVEKET !"
30 FOR I=0 TO 9 : INPUT A$(I) : NEXT I :
   CLS
40 I=RND(10)-1 : J=RND(10)-1 : IF I=J THEN 40
50 IF A$(I)="" OR A$(J)="" THEN 40 ELSE PRINTA$(I),A$(J) :
                                            A$(I)="" : A$(J)=""
60 FOR I=0 TO 9 :
     IF A$(I)="" THEN 70 ELSE 40
70 NEXT I
80 PRINT : STOP
```

```
10 REM ..... 9.3 G Y A K O R L A T
20 CLS :
   DIM X(20) :
   PRINT@5*64,"";
30 FOR I=1 TO 10 : X(I)=RND(99) : PRINT X(I); : NEXT I :
   PRINT :
   FOR I=11 TO 20 : X(I)=RND(99) : PRINT X(I); : NEXT I
40 PRINT : PRINT : PRINT "EGY KIS TURELMET KEREK !" : PRINT : PRINT
50 I=1 : J=2
60 IF X(I)>X(J) THEN X=X(J) : X(J)=X(I) : X(I)=X
70 J=J+1 : IF J<21 THEN 60
80 I=I+1 : J=I+1 : IF I<20 THEN 60
90 FOR I=20 TO 11 STEP-1 : PRINTX(I); : NEXT I :
   PRINT :
   FOR I=10 TO 1 STEP-1 : PRINTX(I); : NEXT I
100 GOTO 100
```

```
10 REM ..... 9.4 G Y A K O R L A T
20 DIM A(10,10)
30 FOR I=0 TO 10 : A(I,0)=1 : A(I,I)=0 : NEXT I
40 FOR I=1 TO 10 : FOR J=1 TO 10 : A(I,J)=A(I-1,J-1)+A(I-1,J) : NEXT J,I
50 CLS
60 FOR I=0 TO 10 :
     FOR J=0 TO 10 :
     IF A(I,J)>0 THEN PRINT@30+I*61+J*6,A(I,J);
70 NEXT J,I
80 GOTO 80
```

```
10 REM ..... 9.5 G Y A K O R L A T
20 DIM A(10,5)
30 CLS :
   PRINT"EZ EGY ELLENORZO !" : PRINT@256,"A TANTARGYAK :
MATEK -1 , FIZIKA - 2 , M . NYELV - 3 , M . IRODALOM - 4 ,
TORTENELEM - 5 , FOLDRAJZ - 6 , KEMIA - 7 , BIOLOGIA - 8 ,
                OROSZ - 9 , ENEK - 10"
40 PRINT@704,"OLVASOL  (K)          IRSZ  (B)"
50 A$=INKEY$ : IF A$="B" THEN60 ELSE IF A$="K" THEN 90 ELSE 50
60 PRINT@832,CHR$(14); : INPUT"A TANTARGY KODSZAMA ";X
70 IF I(X)>5 THEN PRINT"EBBOL A TANTARGYBOL TOBB JEGYET NEM
     IRHATSZ !" : FOR I=1 TO 900 : NEXT : GOTO 30
80 INPUT"AZ OSZTALYZAT ";Y :
   A(X,I(X))=Y : I(X)=I(X)+1 :
   GOTO 30
90 PRINT@832,CHR$(14); : INPUT"A TANTARGY KODSZAMA ";X
100 PRINT@896,"A JEGYEK : ";
110 FOR I=0 TO I(X)-1 : PRINTA(X,I); : NEXT I :
   PRINT@930,"MEGNEZTED ?"
120 IF INKEY$="" THEN 120 ELSE 30
```

```
10 REM ..... 9.6 G Y A K O R L A T
20 DIM A(90) :
   CLS :
   PRINT"LOTTOHUZOK !"
30 FOR I=1 TO 1000 : X=RND(90) : A(X)=A(X)+1 : NEXT I :
   CLS
40 PRINT@64,CHR$(14); : INPUT"MELYIK SZAM ERDEKEL ";X :
   IF X<0 OR X>90 THEN 70
50 PRINT@448,"EZT A SZAMOT ",A(X),"ESETBEN HUZTAM KI !"
60 FOR I=1 TO 400 : NEXT
70 PRINT@64,CHR$(255) : GOTO 40
```

```
10 REM ..... 9.7 G Y A K O R L A T
20 DEFINT A-Z : DIM A(50) :
   CLS :
   PRINT"VELETLENSZAMOK"
30 FOR I=1 TO 1000 : X=RND(50) : A(X)=A(X)+1 : NEXT I :
   CLS
40 PRINT"SZAMOLOK" : J=1
50 FOR I=2 TO 50 :
     IF A(J)<A(I) THEN J=I
60 NEXT I :
   X=47/A(J) :
   CLS
70 FOR I=1 TO 50 : FOR J=1 TO X*A(I) : SET(10+2*I,J) : NEXT J,I
80 GOTO 80
```

```
10 REM ..... 9.8 G Y A K O R L A T
20 DEFINT A-Z :
   CLS
30 INPUT"A SZAM   (   >3   ) ";N : PRINT :
   PRINT"AZ ENNEL KISEBB PRIMSZAMOK :" : PRINT
40 DIM A(N) :
   FOR I=1 TO N : A(I)=I : NEXT I
50 FOR I=2 TO N/2 : FOR J=2*I TO N STEP I : A(J)=0 : NEXT J,I
60 FOR I=1 TO N-1 :
     IF A(I)>0 THEN PRINT A(I);
70 NEXT I :
   PRINT : PRINT :
   CLEAR :
   GOTO 30
```

```
10 REM ..... 9.9 G Y A K O R L A T
20 DIM A(15)
30 CLS :
   PRINT" A 16 SZAM :"
40 FOR I=0 TO 15
50   INPUT A(I) : IF A(I)<32 OR A(I)>191
                   THEN PRINT"UJ SZAMOT KEREK !" : GOTO 50
60 NEXT I :
   CLS
70 FOR I=0 TO 15 :
     PRINT@I*64+20,A(I); : PRINT@I*64+35,CHR$(A(I)); :
   NEXT I
80 GOTO 80
```

```
10 REM ..... 9.10 G Y A K O R L A T
20 CLS :
   INPUT"HANY BETUBOL ALL A SZO ";N : N=N-1
30 DEFSTR A : DIM A(N) :
   A=""
40 PRINT : PRINT"KEREM A SZO BETUIT ! " :
   FOR I=0 TO N : INPUT A(I) : NEXT I
50 I=0 : J=1
60 IF A(I)<A(J) THEN 70 ELSE J=I
70 I=I+1 : IF I=J THEN I=I+1
80 IF I<=N THEN 60
90 IF A(J)>"*" THEN A=A(J)+A : A(J)="*" : GOTO 50
100 CLS :
    PRINT@650,A
110 GOTO 110
```

```
10 REM ..... 9.11 G Y A K O R L A T
20 CLS :
   DEFSTR A,B : DIM A(19)
30 PRINT@0,"A  B E T U K E S Z L E T :"
40 FOR I=0 TO 19 :
     A(I)=CHR$(RND(26)+64) : PRINT@2*64+3*I,A(I) :
   NEXT I
50 PRINT@6*64,"A  S Z O : " : A="" :
   FOR I=1 TO 5 : A=A+A(RND(20)-1) : NEXT I :
   PRINT@8*64,A
60 PRINT@640,"UJ SZO  (U)  UJ BETUKESZLET (B) "
70 B=INKEY$ : IF B="U" THEN 50 ELSE IF B="B" THEN 30
80 GOTO 70
```

```
10 REM ..... 9.12 G Y A K O R L A T
20 DIM A(13,1)
30 FOR I=0 TO 13 : READ A(I,0) : READ A(I,1) : NEXT I
40 CLS :
   FOR I=5 TO 45 STEP 10 :
     FOR J=0 TO 13 :
       PRINT@A(J,0)+I*16,CHR$(A(J,1)); :
   NEXT J,I
50 GOTO 50
60 DATA 5,140,197,140,68,170,131,170,194,170,257,170,70,149,
   135,149,200,149,265,149,195,136,196,136,198,132,199,132
```

```
10 REM ..... 9.13 G Y A K O R L A T
20 CLEAR 500 :
   DEFSTR A,C : DIM A(8,8) :
   J=0
30 CLS :
   PRINT@530,"M E M O R I A -JATEK" :
   FOR I=1 TO 500 : NEXT :
   CLS
40 FOR I=1 TO 8 : PRINT@10+I*3,I : PRINT@I*64+72,I : NEXT I
50 PRINT@832,"EGY KIS TURELMET KEREK !" : C=CHR$(127)
60 FOR I=1 TO 8 :
     FOR J=1 TO 8
70   X=RND(63)+128 : IF A(I,J)>"" THEN 90 ELSE A(I,J)=CHR$(X)
80   M=RND(8) : N=RND(8) : IF A(M,N)>"" THEN 80 ELSE A(M,N)=CHR$(X)
90 NEXT J,I
100 FOR I=1 TO 8 : FOR J=1 TO 8 : PRINT@I*64+75+J*3,C; : NEXT J,I
110 PRINT@832,CHR$(14); : INPUT"AZ ELSO LAP KOORDINATAI ";X,Y :
    PRINT@Y*64+75+X*3,A(X,Y);
120 PRINT@896,CHR$(14);:INPUT"A MASODIK LAP KOORDINATAI ";M,N:
    PRINT@N*64+75+M*3,A(M,N); : PRINT@945,"MEGNEZTED ?";
130 IF INKEY$="" THEN 130 ELSE PRINT@832,CHR$(254); :
                               PRINT@896,CHR$(254);
140 IF A(X,Y)=A(M,N) THEN 110 ELSE
    PRINT@Y*64+75+X*3,C; : PRINT@N*64+75+M*3,C; : GOTO 110
```

```
10 REM ..... 9.14 G Y A K O R L A T
20 CLEAR 500 :
   DEFSTR A,B,C : DIM A(8,8) :
   C=CHR$(127) : B=CHR$(167)
30 CLS :
   PRINT@536,"T O R P E D O" :
   FOR I=1 TO 500 : NEXT :
   CLS
40 FOR I=1 TO 8 : PRINT@10+I*3,I : PRINT@I*64+72,I : NEXT I
50 FOR X=1 TO 5
60   I=RND(8) : J=RND(8) : IF A(I,J)>"" THEN 60 ELSE A(I,J)=B
70 NEXT X
80 FOR I=1 TO 8 : FOR J=1 TO 8 : PRINT@I*64+75+J*3,C; : NEXT J,I
90 PRINT@832,CHR$(14); : INPUT"A LOVES KOORDINATAI   ";X,Y :
   IF X<1 OR Y<1 OR X>8 OR Y>8 THEN PRINT@832,CHR$(250); : GOTO 90
100 PRINT@Y*64+75+X*3,A(X,Y); : L=L+1 :
    IF A(X,Y)>"" THEN PRINT@896,"TALALT !" : Z=Z+1 :
    FOR I=1 TO 500 : NEXT : PRINT@896,CHR$(210); :
    IF Z=5 THEN 130 ELSE PRINT@832,CHR$(250); : GOTO 90
110 PRINT@Y*64+75+X*3,"X"; : PRINT@832,CHR$(250);
120 GOTO 90
130 PRINT@960,"A LOVESEID SZAMA  : ";L;
140 GOTO 140
```

```
10 REM ..... 9.15 G Y A K O R L A T
20 DEFINT A-Z : DIM A(90) :
   PRINT"LOTTOHUZOK !"
30 FOR I=1 TO 1000 : X=RND(90) : A(X)=A(X)+1 : NEXT I
40 J=0 :
   FOR I=2 TO 90 :
     IF A(J)<A(I) THEN J=I
50 NEXT I :
   X=47/A(J)
60 CLS :
   PRINT"     GRAFIKUS SZEMLELTETES          ( 1 )
   SZAMSZERU EREDMENY              ( 2 )"
70 D$=INKEY$ : IF D$="1" THEN 80 ELSE IF D$="2" THEN 110 ELSE 70
80 CLS
90 FOR I=1 TO 90 : FOR J=0 TO X*A(I) : SET(I+20,J) : NEXTJ,I
100 D$=INKEY$ : GOTO 70
110 CLS :
    PRINT"MELYIK ERTEKRE VAGY KIVANCSI "; : INPUT N
120 CLS :
    PRINT N, "EZT A SZAMOT " : PRINT : PRINT A(N),"ESETBEN HUZTAM KI
    D$=INKEY$ : GOTO 70
```

```
10 REM ..... 9.16 G Y A K O R L A T
20 CLEAR 1000 :
   DEFSTR A :
   CLS :
   PRINT"KEREM A TIZ SZOT !"
30 FOR I=0 TO 9 : INPUT A(I) : NEXT I :
   PRINT
40 J=0 :
   FOR I=1 TO 9 :
     IF LEN(A(J))<LEN(A(I)) THEN J=I
50 NEXT I :
   IF A(J)="" THEN PRINT : STOP ELSE PRINT A(J), : A(J)="" : GOTO 40
```

```
10 REM ..... 9.17 G Y A K O R L A T
20 DEFSTR A : DIM A(15) :
   CLS
30 FOR I=0 TO 15 : PRINT I+1;".  SZAM ="; : INPUT A(I) : NEXT I
40 CLS :
   FOR I=0 TO 15 : PRINT@I*64+40-LEN(A(I)),A(I); : NEXT I
50 GOTO 50
```

```
10 REM ..... 9.18 G Y A K O R L A T
20 CLEAR 500 :
   DEFSTR A,B :
   CLS
30 PRINT"KEREK 10 DARAB J-T , VAGY LY -T TARTALMAZO SZOT !"
40 FOR I=0 TO 9 : INPUT A(I) : NEXT I :
   CLS
50 PRINT"EGESZITSD KI A HIANYOS SZOT J-VEL VAGY LY-NAL !"
60 X=RND(10)-1 : A=A(X) : I=0 : H=LEN(A)
70 I=I+1 :
   IF MID$(A,I,1)="J" THEN J=1 ELSE
   IF MID$(A,I,2)="LY" THEN J=2 ELSE 70
80 PRINT@530,LEFT$(A,I-1); : PRINT@535,RIGHT$(A,H-I-J+1); :
   PRINT@594+I,""; : INPUT B : PRINT@530,CHR$(31)
90 IF B="J" AND J=1 OR B="LY" AND J=2 THEN PRINT@770,"BRAVO" :
   PRINT@530,A : FOR I=1 TO 500 : NEXT : PRINT@128,CHR$(31) : GOTO 60
100 PRINT@770,"IRGUM - BURGUM !   " : I=0 :
    GOTO 70
```

Tizedik fejezet

```
10 REM ..... 10.1 G Y A K O R L A T
20  CLS
30  INPUT"N= ";N :
    DIM A(N) :
    FOR I=0 TO N-1 :  INPUT A(I) : NEXT
40  GOSUB 60 :
    PRINT :
    FOR I=0 TO N-1 : PRINT A(I); : NEXT
50  PRINT :
    STOP
60  REM
70  J=0 :
    FOR I=0 TO N-1 :
      IF A(I)<A(I+1) THEN J=1 : Z=A(I) : A(I)=A(I+1) : A(I+1)=Z
80  NEXT I :
    IF J=1 THEN 70
90  RETURN
```

```
10 REM ..... 10.2 G Y A K O R L A T
20  DIM B(12),K(12)
30  CLS :
    PRINT@330,"   B E V E T E L   (1)" : PRINT@522,
    "   K I A D A S     (2)" :
    PRINT@714,"   O S S Z E S I T E S    (3)"
40  A$=INKEY$ :
    IF A$="1" THEN GOSUB 60 ELSE IF A$="2" THEN
    GOSUB 80 ELSE IF A$="3" THEN GOSUB 100 ELSE 40
50  GOTO 30
60  REM BEVETEL
70  CLS :
    INPUT"HANYADIK HONAP ";X : PRINT : INPUT"A BEVETEL ";Y :
    B(X)=B(X)+Y :
    RETURN
80  REM KIADAS
90  CLS :
    INPUT"HANYADIK HONAP ";X : PRINT : INPUT"A KIADAS ";Y :
    K(X)=K(X)+Y :
    RETURN
100 REM OSSZESITES
110 CLS :
    PRINT@2*64,"BEVETELEK :" :
    FOR I=1 TO 12 : PRINTB(I); : B=B+B(I) : NEXT :
    PRINT@6*64,"KIADASOK :" :
    FOR I=1 TO 12 : PRINTK(I); : K=K+K(I) : NEXT
120 PRINT@10*64,"AZ OSSZES BEVETEL :";B :
    PRINT@12*64,"AZ OSSZES KIADAS :";K :
    PRINT@14*64,"AZ EGYENLEG :";B-K; :
    PRINT@15*64,"ELOLVASTAD ?"
130 IF INKEY$="" THEN 130 ELSE RETURN
```

```
10 REM ..... 10.3 G Y A K O R L A T
20  CLS :
    DEFINT A-Z :
    FOR I=1 TO 3 : X(I)=RND(119) : Y(I)=RND(42) : NEXT
30  I=RND(3) : X=X(I) : Y=Y(I) : GOSUB 70
40  Z=RND(4) :
    IF Z=1 THEN X=X-8 ELSE IF Z=2 THEN X=X+8 ELSE
    IF Z=3 THEN Y=Y-4 ELSE Y=Y+4
50  IF X<0 THEN X=0 ELSE IF Y<0 THEN Y=0 ELSE
    IF X>119 THEN X=119 ELSE IF Y>42 THEN Y=42
60  GOSUB 100 : X(I)=X : Y(I)=Y : GOTO 30
70  REM TOROL
80  FOR J=0 TO 4 :
      RESET(X+2*J,Y) : RESET(X,Y+J) :
      RESET(X+8-2*J,Y+4) : RESET(X+8,Y+4-J) :
    NEXT
90  RETURN
100 REM RAJZOL
110 FOR J=4 TO 0 STEP -1 :
      SET(X+2*J,Y) : SET(X,Y+J) :
      SET(X+8,Y+4-J) : SET(X+8-2*J,Y+4) :
    NEXT
120 RETURN
```

```
10 REM ..... 10.4 G Y A K O R L A T
20  CLS :
    DEFINT A-Z
30  FOR I=1 TO 10 :
      FOR J=1 TO 5 :
        SET(20*J,I) : SET(20*J+10,I) :
        SET(20*J+I,1) : SET(20*J+I,10) :
    NEXT J,I
40  FOR I=1 TO 5 : X(I)=10*I+2 : NEXT
50  I=RND(5) : X=X(I) : IF RND(2)-1 THEN GOSUB 70 ELSE GOSUB 100
60  GOTO 50
70  REM BECSUK
80  FOR I=X-1 TO X : FOR J=1 TO 3 : PRINT@J*64+I,CHR$(127); : NEXT J,I
90  RETURN
100 REM KINYIT
110 FOR I=X TO X-1 STEP-1 : FOR J=1 TO 3 : PRINT@J*64+I," "; : NEXT J,I
120 RETURN
```

```
10 REM ..... 10.5 G Y A K O R L A T
20 CLS
30 H=-1 : GOSUB 50 : P=-1 : GOSUB 80 : S=-1 : GOSUB 110
40 GOSUB 110 : GOTO 40
50 REM ORA
60 H=H+1 : IF H>23 THEN H=0
70 PRINT@5*64+35,H,"ORA"; : RETURN
80 REM PERC
90 P=P+1 : IF P>59 THEN P=0 : GOSUB 50
100 PRINT@8*64+35,P,"PERC"; :
    RETURN
110 REM MASODPERC
120 S=S+1 : IF S>59 THEN S=0 : GOSUB 80
130 PRINT@11*64+35,S,"MASODPERC"; :
    RETURN
```

```
10 REM ..... 10.6 G Y A K O R L A T
20 X=0 : Y=10
30 X=X+2 : IF X>57 THEN X=0
40 CLS :
   GOSUB 50 : GOTO 30
50 REM BABU
60 PRINT@320+X," ***** " : PRINT@384+X,"* A A *" :
   PRINT@448+X,"*  T  *" : PRINT@512+X,"*  V  *" :
   PRINT@576+X,"  ***"
70 FOR I=1 TO 30 : NEXT
80 RETURN
```

```
10 REM ..... 10.7 G Y A K O R L A T
20  DIM A(5,5)
30  CLS :
    GOSUB 110
40  FOR J=0 TO 5 :
      FOR I=0 TO 5 : X(I)=A(J,I) : NEXT I :
      GOSUB 170 : Y(J)=X :
    NEXT J
50  PRINT@100,""; :
    FOR I=0 TO 5 : PRINTY(I); : NEXT
60  FOR J=0 TO 5 : X(J)=Y(J) : NEXT :
    GOSUB 140 :
    PRINT@640,"M I N I M A X :";X
70  FOR I=0 TO 5 :
      FOR J=0 TO 5 : X(J)=A(J,I) : NEXT J :
      GOSUB 140 : Y(I)=X :
    NEXT I
80  PRINT@164,""; :
    FOR I=0 TO 5 : PRINTY(I); : NEXT
90  FOR I=0 TO 5 : X(I)=Y(I) : NEXT :
    GOSUB 170 :
    PRINT@768,"M A X I M I N :";X
100 IF INKEY$="" THEN 100 ELSE 30
110 REM FELTOLTES
120 FOR I=0 TO 5 :
      FOR J=0 TO 5 : A(I,J)=RND(9) : PRINT@I*64+2*J,A(I,J); :
    NEXT J,I
130 RETURN
140 REM MAXI
150 X=0 :
    FOR K=0 TO 5 :
      IF X(K)>X THEN X=X(K)
160 NEXT :
    RETURN
170 REM MINI
180 X=10 :
    FOR K=0 TO 5 :
      IF X(K)<X THEN X=X(K)
190 NEXT :
    RETURN
```

```
10 REM ..... 10.8 G Y A K O R L A T
20 CLS
30 FOR I=1 TO 3 :
     PRINTI; : INPUT".   PONT KOORDINATAI ";X(I),Y(I) :
     IF X(I)<0 OR Y(I)<0 OR X(I)>127 OR Y(I)>47 THEN 30
40 NEXT :
   X(0)=X(3) : Y(0)=Y(3) :
   CLS
50 FOR O=0 TO 2 :
     X=X(O) : Y=Y(O) : A=X(O+1) : B=Y(O+1) :
     GOSUB 70 :
   NEXT O
60 IF INKEY$="" THEN 60 ELSE 20
70 REM SZAKASZ RAJZOLO
80 IF ABS(X-A)<ABS(Y-B) THEN 90 ELSE 130
90 IF Y>B THEN S=-1 ELSE S=1
100 FOR I=Y TO B STEP S :
      IF X=A THEN SET(X,I) : GOTO 120
110   J=X-(X-A)*(Y-I)/(Y-B) : SET(J,I)
120 NEXT I :
    RETURN
130 IF X>A THEN S=-1 ELSE S=1
140 FOR I=X TO A STEP S :
      IF Y=B THEN SET(I,Y) : GOTO 160
150   J=Y-(Y-B)*(X-I)/(X-A) : SET(I,J)
160 NEXT :
    RETURN
```

```
10 REM ..... 10.9 G Y A K O R L A T
20    CLS
30    DEFINT A-Z :
      PRINT: INPUT"A SZAMOK HANYJEGYUEK ";N :
      DIM A(N),B(N),E(N)
40    PRINT"AZ EGYIK SZAM SZAMJEGYEI ?" :
      FOR I=1 TO N  : INPUT A(I) : NEXT :
      PRINT"A MASIK SZAM SZAMJEGYEI ?" :
      FOR I=1 TO N  : INPUT B(I) : NEXT
50    FOR I=N TO 0 STEP-1 : F=A(I)+B(I)+C : GOSUB 80 : NEXT
60    CLS :
      PRINT"AZ EREDMENY :"; :
      IF E(0)=0 THEN J=1 ELSE J=0
70    FOR I=J TO N : PRINT E(I); : NEXT :
      PRINT :
      CLEAR :
      GOTO 30
80    REM  MODULO 10
90    IF F>9 THEN E(I)=F-10 : C=1 ELSE E(I)=F : C=0
100   RETURN
```

```
10 REM ..... 10.10 G Y A K O R L A T
20    CLS
30    DEFINT A-Z :
      INPUT"A SZAMOK HANY JEGYUEK ";N :
      DIM A(N),B(N),E(2*N)
40    PRINT"AZ EGYIK SZAM SZAMJEGYEI ?" :
      FOR I=1 TO N : INPUT A(I) : NEXT :
      PRINT"A MASIK SZAM SZAMJEGYEI ?" :
      FOR I=1 TO N : INPUT B(I) : NEXT
50    F=A(N)*B(N) : K=2*N :
      GOSUB 130
60    FOR K=2*N-1  TO 2 STEP -1 :
        V=1 :
        IF K<N+1 THEN V=N+2-K
70      Z=N :
        IF K>N+1 THEN Z=2*N-K+1
80      X=V+K-N-1 : Y=N+1-V : F=0
90      FOR I=V TO Z : F=F+A(X)*B(Y) : X=X+1 : Y=Y-1 : NEXT I
100     F=F+C :
        GOSUB 130
110   NEXT K :
      E(1)=C
120   CLS :
      FOR I=1 TO N : PRINT@2*I,A(I);: PRINT@64+2*I,B(I);: NEXT :
      FORI=1 TO 2*N : PRINT@256+2*I,E(I); : NEXT :
      PRINT : PRINT :
      CLEAR :
      GOTO 30
130   REM MOD 10
140   C=F/10 : E(K)=F-10*C
150   RETURN
```

```
10 REM ..... 10.11 G Y A K O R L A T
20    DEFINTA-Z : DIM  H(12) :
      FOR I=1 TO 12 : READ H(I) : NEXT :
      CLS
30    N=0 :
      PRINT"A MAI DATUM ?" : INPUT"EV , HO , NAP ";X,Y,Z:
      PRINT"A SZULETESI DATUM ?" : INPUT"EV , HO , NAP ";A,B,C
40    IF X<A THEN 30 ELSE IF X=A THEN IF Y<B THEN 30 ELSE
      IF Y=B THEN IF Z<C THEN 30
50    IF X=A AND Y=B THEN N=Z-C : GOTO 150
60    IF X>A THEN 90
70    K=C : GOSUB 190 : K=0
80    B=B+1 : IF B<Y THEN GOSUB 190 : GOTO 80 ELSE N=N+Z : GOTO 150
90    K=C : GOSUB 190 : K=0
100   B=B+1 : IF B>12 THEN 110 ELSE GOSUB 190 : GOTO 100
110   A=A+1 : IF A<X  THEN GOSUB 220 : GOTO 110
120   B=0
130   B=B+1 : IF B<Y THEN GOSUB 190 GOTO 130
140   N=N+Z
150   PRINT@9*64,"AZ ELTELT NAPOK SZAMA : ",N : PRINT : PRINT :
      STOP
160   REM SZOKOEV
170   I=A/4 : IF I*4=A THEN I=0 ELSE I=1
180   RETURN
190   REM HONAP
200   IF B=2 THEN J=A : GOSUB 160 : IF I=0 THEN N=N+29-K :
      RETURN ELSE N=N+28-K : RETURN
210   N=N+H(B)-K :
      RETURN
220   REM EV
230   GOSUB 160 : IF I=0 THEN N=N+366 ELSE N=N+365
240   RETURN
250   DATA 31,28,31,30,31,30,31,31,30,31,30,31
```

Tizenegyedik fejezet

```
10 REM ..... 11.1 G Y A K O R L A T
20    DEFINT A-Z :
      X=15360 : I=16537 : Z=PEEK(X) : POKE X,95
30    Y=PEEK(I) :
      IF Y=0 THEN 30 ELSE POKE I,0 : IF Y=10 THEN
      POKE X,Z : X=X+64 ELSE IF Y=91 THEN POKE X,Z : X=X-64 ELSE
      IF Y=8 THEN POKE X,Z : X=X-1 ELSE IF Y=9 THEN POKE X,Z :
      X=X+1 ELSE 30
40    IF X>16383 THEN X=16383 ELSE IF X<15360 THEN X=15360
50    Z=PEEK(X) : POKE X,95 :
      GOTO 30
```

```
10 REM ..... 11.2 G Y A K O R L A T
20    DEFINT A-Z :
      X=15360 : I=16537 : Z=PEEK(X) : POKE X,95
30    Y=PEEK(I) :
      IF Y=0 THEN 30 ELSE POKE I,0 : IF Y=10 THEN
      POKE X,Z : X=X+64 ELSE IF Y=91 THEN POKE X,Z : X=X-64 ELSE
      IF Y=8 THEN POKE X,Z : X=X-1 ELSE IF Y=9 THEN POKE X,Z :
      X=X+1 ELSE 30
40    IF X<15359 THEN X=X+1024 ELSE IF X=15359 THEN X=15423ELSE
      IF X=16384 THEN X=16320 ELSE IF X>16384 THEN X=X-1024
50    Z=PEEK(X) : POKE X,95 :
      GOTO 30
```

```
10 REM ..... 11.3 G Y A K O R L A T
20    REM RAJZOLO PROGRAM
      NEW LINE   billentyuvel karaktermodvaltas !
30    DEFINT A-Z :
      CLS :
      X=15360 : I=16537 : Y=32 : Z=31 : Y1=1:
      GOTO 110
40    Y=PEEK(I) :
      IF Y=0 THEN 40 ELSE POKE I,0 : IF Y=10 THEN
      POKE X,Z : X=X+64 ELSE IF Y=91 THEN POKEX,Z : X=X-64 ELSE
      IF Y=8 THEN POKE X,Z : X=X-1 ELSE IF Y=9 THEN POKE X,Z :
      X=X+1 ELSE 70
50    IF X>16383 THEN X=16383 ELSE IF X<15360 THEN X=15360
60    Z=PEEK(X) : POKE X,95 :
      GOTO 40
70    RETURN
80    POKE X,Y : Z=Y :
      RETURN
90    IF Y<48 AND Y>41 THEN Z=Y+49 ELSE IF Y<91 THEN Z=Y+75 ELSE
      Z=Y+69
100   POKE X,Z :
      RETURN
110   GOSUB 40 :
      IF Y=13  THEN IF Y1=1 THEN Y1=0 : GOTO 110
               ELSE Y1=1 : GOTO 110
120   IF Y1=1 THEN GOSUB 80 ELSE GOSUB 90
130   GOTO 110

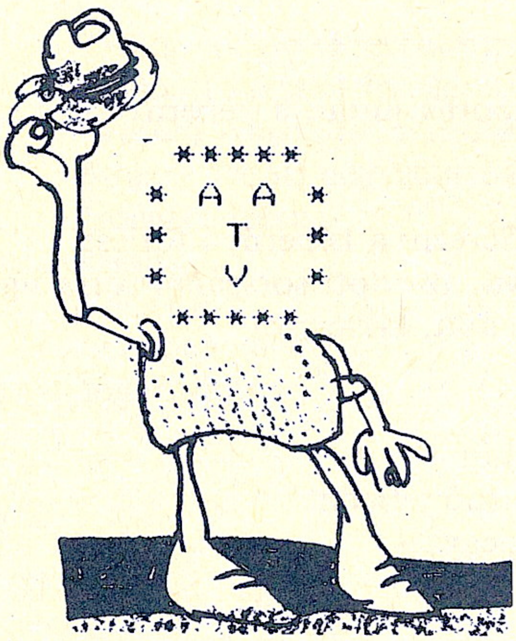

10 REM ..... 11.4 G Y A K O R L A T
20    CLS :
      X=15835 : X1=X :  Y=15843 : I=16537 : L=0
30    FOR J=0 TO 4 : POKE X+2*J,RND(26)+64 : NEXT :
      PRINT@10,".  CSERE";
40    Z=PEEK(X) : POKE X,95
50    K=PEEK(I) :
      IF K=0 THEN 50 ELSE POKE I,0 : IF K=8 THEN POKE X,Z :
      X=X-2 ELSE IF K=9 THEN POKE X,Z : X=X+2 ELSE 80
60    IF X<X1 THEN X=X1 ELSE IF X>Y THEN X=Y
70    Z=PEEK(X) : POKE X,95 :
      GOTO 50
80    L=L+1 :
      PRINT@0,L; :
      POKE X,K : Z=K :
      GOTO 50


10 REM ..... 11.5 G Y A K O R L A T
20 DEFINT H-L
30 OUT 31,7 : OUT 30,254
40 OUT 31,8 : OUT 30,15
50 OUT 31,0
60 RESTORE
70 READ HANG : IF HANG=999 THEN 120
80 READ IDO
90 OUT 30,HANG
100 FOR J=0 TO 33*IDO : NEXT J
110 GOTO 70
120 OUT 30,0
130 STOP
140 REM A HANG ADATAI
150 DATA 124,11,999
```

```
10 REM ..... 11.6 G Y A K O R L A T
20 DEFINT H-L
30 OUT 31,7 : OUT 30,254
40 OUT 31,8 : OUT 30,15
50 OUT 31,0
60 RESTORE
70 READ HANG : IF HANG=999 THEN 120
80 READ IDO
90 OUT 30,HANG
100 FOR J=0 TO 99*IDO : NEXT J
110 GOTO 70
120 OUT 30,0
130 STOP
140 REM ROZSA SANDOR BALLADAJA
150 DATA 104,2,0,0,104,2,0,0,117,2,0,0,117,2,0,0,104,2,
    0,0,104,2,0,0,117,2,0,0,104,2,0,0
160 DATA 92,2,0,0,117,4,104,2,0,0,104,4,0,0,0,4
170 DATA 139,2,0,0,139,2,0,0,139,2,0,0,139,2,0,0,117,2,0,0,
    139,2,0,0,139,2,0,0,155,1,165,1,0,0
180 DATA 155,4,0,0,165,3,155,1,0,0,139,4,0,0,0,4
190 DATA 208,4,0,0,208,3,222,1,0,0,208,4,0,0,222,4,0,0
200 DATA 165,2,0,0,165,2,0,0,155,2,0,0,155,2,0,0,139,2,0,0,
    124,2,0,0,110,4,0,0
210 DATA 104,2,0,0,110,1,139,1,0,0,139,2,0,0,139,1,155,1,0,0,
    155,2,0,0,155,1,165,1,0,0,165,1,185,1,0,0,222,2,0,0
220 DATA 208,2,0,0,208,6,0,0,208,4,0,0,0,4
230 DATA 999
```

```
10 REM ..... 11.7 G Y A K O R L A T
20 CLS
30 CLEAR 120
40 OUT 31,7 : OUT 30,254 : OUT 31,8 : OUT 30,15
50 D=7
60 S$=STRING$(57,CHR$(191)) :
   PRINT@64*8,S$; : PRINT@64*12,S$;
70 FOR I=0 TO 56 STEP 7
80   FOR J=9 TO 11 : PRINT@64*J+I,CHR$(191); : NEXT J
90 NEXT I
100 RESTORE
110 FOR I=0 TO D : READ T$,T : PRINT@64*10+7*I+4%,T$; : NEXT I
120 I=D
130 PRINT@64*10+7*I+4," ";
140 PRINT@64*10+7*I+4,T$;
150 H$=INKEY$ : IF H$="" THEN 130
160 RESTORE
170 FOR I=0 TO D
180   READ T$,T
190   IF H$=T$ THEN OUT 31,0 : OUT 30,T : GOTO 130
200 NEXT I
210 PRINT@64*15,CHR$(30);"ISMERETLEN HANG !!! ";
220 PRINT"A FOLYTATASHOZ USS LE EGY BILLENTYUT !!";
230 OUT 31,0 : OUT 30,0
240 PRINT@14*64,""; :
    A$=INKEY$ :
    IF A$="" THEN 240 ELSE PRINT@64*15,CHR$(31); : GOTO 120
250 STOP
260 DATA C,208,D,185,E,165,F,155,G,139,A,124,H,110,c,104
```

```
10 REM ..... 11.8 G Y A K O R L A T
20 OUT 31,7 : OUT 30,254
30 OUT 31,8 : OUT 30,15
40 OUT 31,0
50 X=RND(63)+131
60 IF RND(2)-1 THEN J=15360 : L=16383 : S=RND(100)
            ELSE J=16383 : L=15360 : S=-RND(100)
70 CLS
80 FOR M=J TO L STEP S
90   POKE M,X
100    OUT 30,(M-15360)/5
110 NEXT
120 GOTO 50
```



A BASIC-KULCSSZAVAK TÁRGYMUTATÓJA ÉS MAGYAR ÉRTELMEZÉSE

Első fejezet

SET	valamely állapotba juttat, a képernyő egy grafikus pontját kigyújtja
RUN	fut, indul
CLS	Clear of the Screen: a képernyő törlése
GOTO	megy valahova, az adott sorszámú sorra ugrik
RESET	alaphelyzetbe állít, kikapcsol
NEW	új
LET	legyen
PRINT	(ki)nyomtat, kiír
INPUT	bemenet, bemenő jel/adat
REM	Remark: megjegyzés

Második fejezet

IF	ha
THEN	akkor
ELSE	különben
STOP	állj!
END	vége
CONT	Continue: folytat
FOR TO	-tól, -ig
STEP	lépés
NEXT	következő

Harmadik fejezet

DEFINT	Define Integer: egész típus definiálása
POINT	pont
AND	és
OR	vagy

Negyedik fejezet ———

Ötödik fejezet

DEFSNG	Define Single (Precision): egyszeres pontosság definiálása
DEFDBL	Define Double (Precision): dupla pontosság definiálása
ABS	(Value) Absolute: abszolútérték
SQR	Square Root: négyzetgyök
INT	Integer: egész
SIN	szinusz
COS	koszinusz
EXP	exponenciális
LOG	logaritmus

Hatodik fejezet

RND	Random: véletlen
DEFSTR	Define String: karaktertípus definiálása
INKEY	In Keyboard: billentyűzeten

Hetedik fejezet

CLEAR	töröl
STRING	(karakter)lánc
CHR	Character: karakter
ASC	ASCII
LEN	Length: hosszúság, valaminek a hossza
LEFT	bal
RIGHT	jobb
MID	közép(en levő)

VAL	Value: érték
STR	String

Nyolcadik fejezet

TAB	tabulálás
DATA	adat
READ	olvas
RESTORE	visszaállít

Kilencedik fejezet

DIM	Dimension: dimenzió

Tizedik fejezet

GOSUB	szubrutin hívás
RETURN	visszatérés
ON	-nál, -nél

Tizenegyedik fejezet

PEEK	megnéz, kukucskál
POKE	zsák
OUT	ki, kifelé

Az ASCII-kódok táblázata

32	SPACE	48	Ø	64	@	80	P	96		112	p	128	SPACE	144		160		176	
33	!	49	1	65	A	81	Q	97	a	113	q	129		145		161		177	
34	"	50	2	66	B	82	R	98	b	114	r	130		146		162		178	
35	#	51	3	67	C	83	S	99	c	115	s	131		147		163		179	
36	$	52	4	68	D	84	T	100	d	116	t	132		148		164		180	
37	%	53	5	69	E	85	U	101	e	117	u	133		149		165		181	
38	&	54	6	70	F	86	V	102	f	118	v	134		150		166		182	
39	'	55	7	71	G	87	W	103	g	119	w	135		151		167		183	
40	(	56	8	72	H	88	X	104	h	120	x	136		152		168		184	
41	)	57	9	73	I	89	Y	105	i	121	y	137		153		169		185	
42	*	58	:	74	J	90	Z	106	j	122	z	138		154		170		186	
43	+	59	;	75	K	91	[	107	k	123	{	139		155		171		187	
44	,	60	<	76	L	92	\	108	l	124	¦	140		156		172		188	
45	–	61	=	77	M	93	]	109	m	125	}	141		157		173		189	
46	.	62	>	78	N	94	∧	110	n	126	~	142		158		174		190	
47	/	63	?	79	O	95	_	111	o	127	⁘	143		159		175		191	

Megjegyzések:

- Géptípusonként bizonyos ASCII-kódokhoz tartozó karakterekben eltérések lehetnek.
- Érdekességképpen megemlítjük, hogy az ASCII-kódokat nem minden személyi számítógép használja, léteznek más kódrendszerek is.
- A Ø–31 közötti ASCII-kódok közül néhánynak a jelentésére utaltunk a hetedik fejezetben (ún. vezérlő karakterek).

Tartalom

Tankönyvkiadó Vállalat. A kiadásért felelős: Petró András igazgató. Dabasi Nyomda (85-1249), Budapest–Dabas, 1986. Felelős vezető: Bálint Csaba igazgató. Raktári szám: 53 023. Szedte a Nyomdaipari Fényszedő Üzem (85.7713/8). Felelős szerkesztő: Nagy Józsefné. Műszaki igazgatóhelyettes: Lojd Lajos. Grafikai szerkesztő: Démuth Ágnes. Műszaki szerkesztő: Gregor László. A kézirat nyomdába érkezett: 1985. június. Megjelent: 1986. március. Példányszám: 21 800. Terjedelem: 14,30 (A/5) ív. Készült: fényszedéssel, íves ofszetnyomással, az MSZ 5601–59 és az MSZ 5602–55 szabvány szerint.

43,— Ft

53 023